

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1 9 9 7 年 1 月 1 4 日

出 願 番 号

Application Number:

平成 9 年特許願第 0 1 7 3 3 3 号

出 願 人

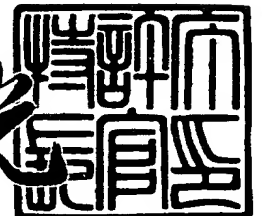
Applicant (s):

ヤマハ株式会社

1 9 9 7 年 6 月 1 3 日

特 許 庁 長 官
Commissioner,
Patent Office

荒井 寿光



出証番号 出証特平 0 9 - 3 0 4 5 7 2 3

【書類名】 特許願

【整理番号】 YC26270

【提出日】 平成 9年 1月14日

【あて先】 特許庁長官 殿

【国際特許分類】 G10H 7/00

【発明の名称】 音源波形データ生成方法および装置

【請求項の数】 6

【発明者】

【住所又は居所】 静岡県浜松市中沢町10番1号 ヤマハ株式会社内

【氏名】 増田 英之

【発明者】

【住所又は居所】 静岡県浜松市中沢町10番1号 ヤマハ株式会社内

【氏名】 磯崎 善政

【発明者】

【住所又は居所】 静岡県浜松市中沢町10番1号 ヤマハ株式会社内

【氏名】 鈴木 秀雄

【発明者】

【住所又は居所】 静岡県浜松市中沢町10番1号 ヤマハ株式会社内

【氏名】 平野 正志

【特許出願人】

【識別番号】 000004075

【氏名又は名称】 ヤマハ株式会社

【代表者】 上島 清介

【代理人】

【識別番号】 100102635

【弁理士】

【氏名又は名称】 浅見 保男

【代理人】

【識別番号】 100106459

【弁理士】

【氏名又は名称】 高橋 英生

【代理人】

【識別番号】 100086841

【弁理士】

【氏名又は名称】 脇 篤夫

【先の出願に基づく優先権主張】

【出願番号】 平成 8年特許願第248592号

【出願日】 平成 8年 8月30日

【手数料の表示】

【予納台帳番号】 037338

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9305279

【包括委任状番号】 9507139

【書類名】 明細書

【発明の名称】 音源波形データ生成方法および装置

【特許請求の範囲】

【請求項1】 波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした音源波形データ生成方法において、

上記波形演算アルゴリズムを実行する演算部の負荷状態に応じて、生成される前記音源波形データのサンプリング周波数が変更されるようにしたことを特徴とする音源波形データ生成方法。

【請求項2】 波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした音源波形データ生成方法において、

実行される上記波形演算アルゴリズムのうち、ゲインが実質的にゼロとされている部分の演算を省略するようにしたことを特徴とする音源波形データ生成方法。

【請求項3】 波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした演算部を少なくとも備え、

上記演算部の負荷状態に応じて、生成される前記音源波形データのサンプリング周波数が変更されるようにしたことを特徴とする音源波形データ生成装置。

【請求項4】 波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした演算部を少なくとも備え、

上記演算部で実行される上記波形演算アルゴリズムのうち、ゲインが実質的にゼロとされている部分の演算を省略するようにしたことを特徴とする音源波形データ生成装置。

【請求項5】 波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした演算部を少なくとも備え、

上記演算部の負荷状態に応じて、生成される前記音源波形データのサンプリング周波数が変更されるようになされており、

前記演算部には、信号サンプルを順次記憶する記憶手段からなる遅延手段が含まれており、

該遅延手段は、入力信号サンプルを前記記憶手段に書き込むライトポインタと

、遅延出力サンプルを前記記憶手段から読み出すリードポイントと、

前記ライトポイントと前記リードポイントとのアドレス差が所望の遅延量になるよう制御するアドレス発生手段とを備え、

該アドレス発生手段は、前記信号サンプルのサンプリング周波数が第1サンプリング周波数とされたときは、前記ライトポイントと前記リードポイントを1アドレスずつ更新制御し、前記信号サンプルのサンプリング周波数が第1サンプリング周波数の $1/n$ の第2サンプリング周波数とされた場合は、前記ライトポイントを1アドレスずつ n 回更新すると共に、前記リードポイントは n アドレスずつ更新制御されることを特徴とする音源波形データ生成装置。

【請求項6】 少なくとも信号遅延処理を含む波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした演算部と、

該演算部がアクセスすることのできる記憶手段を少なくとも備え、

前記演算部は、前記信号遅延処理の実行によって前記記憶手段において波形信号サンプルを順次記憶し、かつ所定の遅延特性をもって遅延出力としての波形信号サンプルを読み出し出力する少なくとも1つの遅延手段を構成し、

該遅延手段は、記憶手段に作成された複数に分割された遅延領域により構成されており、空き状態とされた遅延領域を新規発音用の遅延領域として割り当てると共に、それまでの発音用として割り当てられていた遅延領域を空き状態として待機させるようにしたことを特徴とする音源波形データ生成装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、汎用の演算処理手段を使用して、ソフトウェアとして記述された音源波形データを生成する波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした音源波形データ生成方法および装置に関する。

【0002】

【従来の技術】

最近、ソフトウェアをパーソナルコンピュータ等の汎用処理装置により実行させることにより音源波形データを生成し、生成した音源波形データをCODEC (D

Aコンバータ)においてアナログ楽音波形データに変換して発音するようにすることが試みられている。このようにソフトウェアにより音源波形データを生成する音源をソフトウェア音源と呼んでいる。

また、音源専用LSIやそれに準じた仕様をマイクロプログラムとして実行するデジタルシグナルプロセッサ(DSP)を備える音源専用機や、パーソナルコンピュータに拡張ボードとして装着される音源ボードにより、音源波形データを生成することも行われている。このような音源は、ハードウェア音源と呼ばれている。

【0003】

【発明が解決しようとする課題】

ところで、パーソナルコンピュータでは、一般に複数のアプリケーションソフトを動作させており、ソフトウェア音源のアプリケーションソフトを動作させているときに、例えば、カラオケ用のアプリケーションソフトやゲーム用のアプリケーションソフトを同時に動作させることが考えられる。

すると、パーソナルコンピュータにおける中央処理装置(CPU)の負荷が大きくなり、ソフトウェア音源では音源波形データの生成が遅れ、極端な場合には楽音の発音が途切れるおそれが生じるようになる。また、CPUがマルチタスクを行っている場合には、楽音発生以外のタスク処理が滞ってしまうなどの障害を生じる可能性がある。

そこで、本発明はCPUの負荷が大きい場合であっても楽音の発音が途切れることのない音源波形データを生成できるとともに、CPUがマルチタスクを行っている場合には、他のタスク処理が滞ってしまうことのない音源波形データ生成方法および装置を提供することを目的としている。

【0004】

また、ハードウェア音源においては、波形演算アルゴリズムをDSP等で実行させることにより音源波形データを生成している。演算を実行するDSPの能力は年々向上しているが、従来の音源波形データ生成方法ではその能力を十分発揮することができなかった。

そこで、本発明は、ハードウェア音源において演算手段の能力を十分発揮して

、従来では得ることのできない高精度の波形出力を得ることのできる音源波形データ生成方法および装置を提供することを目的としている。

【0005】

【課題を解決するための手段】

上記目的を達成するために、本発明の音源波形データ生成方法は、波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした音源波形データ生成方法であり、上記波形演算アルゴリズムを実行する演算部の負荷状態に応じて、生成される前記音源波形データのサンプリング周波数を変更するようにしている。

【0006】

これにより、本発明は、CPUの負荷が大きいときはサンプリング周波数を下げることができるようにすることができ、楽音の発音が途切れることのない音源波形データを生成することができる。また、CPUがマルチタスク処理をしている場合においても各タスク処理のCPUパワーを配分する余裕を得ることができる。さらに、CPUの負荷が軽いときは通常のサンプリング周波数より高いサンプリング周波数とすることができ、高精度の音源波形データを生成することができるようになる。

この場合、サンプリング周波数を変更することに代えて、発音数を変更するようにしてもよい。

【0007】

また、本発明の他の音源波形データ生成方法は、波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした音源波形データ生成方法において、実行される上記波形演算アルゴリズムのうち、ゲインが実質的にゼロとされている部分の演算を省略するようにしている。

【0008】

これにより、本発明は、CPUの負荷が極力大きくならないように効率的な演算を行うことができるので、楽音の発音が途切れることのない音源波形データを生成することができる。また、効率的な演算を行えることからサンプリング周波数を従来より高いサンプリング周波数とすることができ、高精度の音源波形デー

タを生成することができるようになる。

【0009】

本発明の音源波形データ生成装置は、上記したそれぞれの音源波形データ生成方法における波形演算アルゴリズムを実行する演算部を少なくとも備えるようにしたものである。

そして、これらの音源波形データ生成装置は、前記した音源波形データ生成方法と同様の作用を奏するものである。

【0010】

さらに、本発明の他の音源波形データ生成装置は、波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした演算部を少なくとも備え、上記演算部の負荷状態に応じて、生成される前記音源波形データのサンプリング周波数を変更されるようになされており、前記演算部には、信号サンプルを順次記憶する記憶手段からなる遅延手段が含まれており、該遅延手段は、入力信号サンプルを前記記憶手段に書き込むライトポインタと、遅延出力サンプルを前記記憶手段から読み出すリードポインタと、前記ライトポインタと前記リードポインタとのアドレス差が所望の遅延量になるよう制御するアドレス発生手段とを備え、該アドレス発生手段は、前記信号サンプルのサンプリング周波数が第1サンプリング周波数とされたときは、前記ライトポインタと前記リードポインタを1アドレスずつ更新制御し、前記信号サンプルのサンプリング周波数が第1サンプリング周波数の $1/n$ の第2サンプリング周波数とされた場合は、前記ライトポインタを1アドレスずつ n 回更新すると共に、前記リードポインタは n アドレスずつ更新制御されるようにしている。

【0011】

このような本発明によれば、波形演算により生成される信号サンプルのサンプリング周波数を変更するようにしても、サンプリング周波数の変更に伴い、遅延手段の遅延制御を行うようにしたので、生成される信号サンプルのサンプリング周波数を変更されても、再生したときに音高が変更されることを防止することができる。

また、遅延手段が効果付与手段内に備えられている場合は、サンプリング周波

数が変更されても効果付与量が変更されることを防止することができ、音色の変化を抑制することができる。

【0012】

さらにまた、本発明の他の音源波形データ生成装置は、少なくとも信号遅延処理を含む波形演算アルゴリズムを実行することにより、音源波形データを生成するようにした演算部と、該演算部がアクセスすることのできる記憶手段を少なくとも備え、前記演算部は、前記信号遅延処理の実行によって前記記憶手段において波形信号サンプルを順次記憶し、かつ所定の遅延特性をもって遅延出力としての波形信号サンプルを読み出し出力する少なくとも1つの遅延手段を構成し、該遅延手段は、記憶手段に作成された複数に分割された遅延領域により構成されており、空き状態とされた遅延領域を新規発音用の遅延領域として割り当てると共に、それまでの発音用として割り当てられていた遅延領域を空き状態として待機させるようにしている。

このような発明によれば、遅延領域のクリア時間分だけ楽音生成が遅れることを防止して、遅延領域のクリアの待ち時間をなくすことができる。従って、新たな楽音を直ちに生成することができるようになる。

【0013】

【発明の実施の形態】

図1に本発明の音源波形データ生成方法が実行されるソフトウェアシステムの構成例を示す。

図1において、MIDI出力部AP S 1はMIDIメッセージを出力する部位であり、鍵盤等の演奏操作子やMIDIメッセージを出力するシーケンサ、あるいは、MIDIメッセージを出力するアプリケーションソフトである。MIDI API I F 1は、MIDIメッセージを授受するオペレーションシステムOSが有している第1のアプリケーションプログラミングインターフェースであり、ソフトウェア音源モジュールS S Mは、オペレーティングシステムOSにドライバとしてインストールされているアプリケーションソフトである。

【0014】

このソフトウェア音源モジュールS S Mは、MIDI出力部AP S 1から出力

されるMIDIメッセージを、インターフェースIF1を介して受け取る。そして、ソフトウェア音源モジュールSSMは、受け取ったMIDIメッセージに基づいて音源波形データの生成処理を実行することにより音源波形データを生成する。生成された音源波形データは、オペレーションシステムOSが有している第2のアプリケーションプログラミングインターフェースIF2（"WAVE out API"）を介してオペレーションシステムOSにより受け取られる。

出力デバイスOUDは、オペレーティングシステムOSにインストールされたドライバモジュールであり、ソフトウェア音源モジュールSSMから生成された音源波形データをインターフェース"WAVE out API"（IF2）を介して受け取り、この音源波形データを外部のハードウェア（CODEC HARDWARE）に対して出力している。

【0015】

この出力デバイスOUDは、例えば、ソフト音源モジュールSSMの演算処理により生成されバッファに格納された音源波形データをダイレクトメモリアクセス（DMA）方式により読み出して、ディジタル・アナログコンバータ（DAC）からなる外部ハードウェアに与えるようにしたソフトウェアである。

また、ソフトウェア音源モジュールSSMは、演算により所定のサンプリング周波数FSの音源波形データ（サンプルデータ）を生成する音源部と、この音源部を駆動するためのMIDI出力ドライバ部を含んでいる。このMIDI出力ドライバ部は、受け取ったMIDIメッセージに応じた音制御パラメータをテーブル等から読み出して音源部に供給している。

【0016】

次に、ソフトウェア音源モジュールSSMの動作タイミングチャートを図2に示す。この図に示すように、ソフトウェア音源モジュールSSMは所定の時間長のフレーム毎に設定されている演算時刻毎に起動される。演算に際しては、その直前のフレーム内で受け取ったMIDIメッセージに基づく音制御パラメータがテーブル等から読み出されてバッファに格納されており、このバッファに格納された各種の音制御パラメータに基づいて音源波形を生成する生成演算処理が実行される。

すなわち、図2に示すように時刻T1から時刻T2までのフレームにおいては3つのMIDIメッセージを受け取り、演算時刻T2になるとソフトウェア音源モジュールSSMが起動されて、受け取った各MIDIメッセージに応じた各種パラメータが読み出されてバッファに格納され、前フレームで発音されていた楽音を継続して発音するための音源波形データと、受け取ったMIDIメッセージに基づいて新たに発音される音源波形データとの生成演算が実行される。

【0017】

この音源波形データの生成演算は、1フレーム分に相当するサンプル数の音源波形データが発音チャンネルごとに生成され、各発音チャンネルについての音源波形データが累算されて波形出力バッファに書き込まれる。そして、波形出力バッファの再生を出力デバイスOUDに予約する。この予約が、生成した音源波形データをソフトウェア音源モジュールSSMから第2のインターフェース“WAVE out API”に出力することに相当する。

出力デバイスOUDは、各フレームごとに、直前のフレームにおいて再生予約された波形出力バッファから1サンプルずつ音源波形データを読み出して外部のハードウェアであるDACに送る。例えば、時刻T1から時刻T2までのフレームで生成された音源波形データが書き込まれて再生予約された波形出力バッファからは、時刻T2から時刻T3までのフレームで音源波形データが読み出される。

読み出された音源波形データは、DACによりアナログの楽音波形信号に変換されてサウンドシステムから発音される。

【0018】

次に、本発明の音源波形データ生成方法を具現化した本発明の音源波形データ生成装置を備える処理装置の一構成例の概略を示すブロック図を図3に示す。

図3に示す処理装置において、メイン制御部としてCPU1が使用され、このCPU1の制御の下で本発明の音源波形データ生成方法が、ソフトウェア音源プログラムによる音源波形データ生成処理として実行される。同時に、その他のアプリケーションプログラム等の処理が並列して実行される。CPU1には内部バスを介してROM（リード・オンリ・メモリ）2、RAM（ランダム・アクセス

・メモリ) 3、ディスプレイインターフェース (DISPLAY I/F) 4、HDD (ハード・ディスク装置)、CD-ROM装置 (CD-ROM drv) 7、内部バスと拡張バス間のデータの受け渡しを行うインターフェース (I/F) 8、パソコン用のユーザインターフェースであるカナキー、数字キー、記号キー等を備えるキーボード (KEY BOARD) 10が接続されている。

【0019】

また、CPU1に内部バス、I/F 8、拡張バスを介して、ハードウェア音源を構成するパーソナルコンピュータの拡張ボードである音源ボード (DSP BOARD) 9、ネットワークとのインターフェース (NETWORK I/F) 11、MIDIのインターフェース (MIDI I/F) 12、DAC 14-2を備えるCODEC 14が接続されている。

ROM2には、CPU1の動作プログラム等が格納されており、RAM3には選択された各種の音制御パラメータが格納されるパラメータバッファ領域、演算生成された楽音波形データが格納される波形出力バッファ領域、受信されたMIDIメッセージおよびその受信時刻が格納される入力バッファ領域、および、CPU1のワークメモリ領域等が設定されている。

【0020】

また、ディスプレイ5とディスプレイインターフェース4は、ユーザが処理装置と対話するための表示部であり、HDD6には、オペレーションシステムOS (例えば、マイクロソフト社製のWindows3.1 (商標)、あるいは、Windows95 (商標)) や、ソフトウェア音源モジュールを実現するプログラムや、"MIDI API" "WAVE API"を実現するその他のアプリケーションプログラム等が格納されている。CD-ROM装置7には、CD-ROM 7-1がセットされて、CD-ROM 7-1に記録されているプログラムやデータが読み出される。これらの読み出されたプログラムやデータはHDD6等に格納される。この場合、CD-ROM 7-1にソフトウェア音源を実現する新たな音源プログラムが記録されていると、容易に音源プログラムのバージョンアップ等を行うことができる。

【0021】

音源ボード9は、音源専用のボードであり、FM合成音源の演算あるいは波形

メモリ音源の演算を実行することによりハードウェア音源を実現しており、演算処理を行うDSP 9-1と、各種バッファが設定されると共に、各種音色パラメータや各種バッファ等が設定されているRAM 9-2から構成されている。ネットワークインターフェース11は、イーサネット (ethernet) 等のLANや電話回線等を介してインターネット等に接続するためのインターフェースであり、ネットワークを通じて音源プログラム等のアプリケーションソフトやデータを受け取ることができる。MIDIインターフェース12は外部MIDI機器とのMIDIメッセージの受け渡しを行ったり、鍵盤等の演奏操作子13からのMIDIイベントが入力されるインターフェースである。この、MIDIインターフェース12を介して入力されたMIDIメッセージは、その内容と時刻とがRAM2の入力バッファ領域に格納される。

【0022】

CODEC 14は、ダイレクトメモリアクセス方式で、RAM3の波形出力バッファ領域に格納されている音源波形データを読み出してサンプルバッファ (SAMPLE BUFFER) 14-1に格納し、所定のサンプリング周波数FS (例えば44.1 kHz) の周期毎にサンプルバッファ14-1から1サンプルずつ読み出してDAC 14-2によりアナログ楽音信号に変換して楽音信号出力 (TONE OUTPUT) として出力する。この楽音信号出力は図示されていないサウンドシステムに入力されて楽音が発音される。

以上の構成はパソコン、ワークステーション等と同等であり、それらの上で本発明の音源波形データ生成方法を実施することができる。

【0023】

次に、CPU1の起動制御に基づくソフト音源モジュールSSMによる、本発明の音源波形生成方法の動作を概略説明すると、アプリケーションプログラムAP S1が起動されると、ソフトウェア音源モジュールSSMに第1のインターフェース”MIDI API”を経由してMIDIメッセージが供給され始める。すると、ソフトウェア音源モジュールSSMのMIDI出力ドライバ部が起動され、各MIDIメッセージに応じた音制御パラメータが設定される。そして、この音制御パラメータが当該MIDIメッセージに割り当てられた発音チャンネル用の音源レジス

タに格納され、前記した図2に示すように演算時刻毎に起動される音源部において所定サンプル数の波形データが演算生成されるのである。

【0024】

次に、図4ないし図6に本発明の音源波形データ生成方法における音源モデルの概念を具体化した構成例をブロック図として示すが、この音源モデルはハードウェアとして実現されているのではなくソフトウェアとして実現されている。

図4ないし図6に示す音源モデルは、管楽器系あるいは弦楽器系をシミュレートした音源モデルであり、以下必要に応じて物理モデル音源という。管楽器系の場合の物理モデル音源は、図4に示すように2個の円筒部の管の継ぎ目の部分にマウスピースを取り付けた管楽器がシミュレートされており、弦楽器系の場合には弦の両端がブリッジで固定された撥弦あるいは擦弦楽器がシミュレートされている。

【0025】

図4に示す物理モデル音源は図示するようにループ状とされており、ループ内の全体の遅延時間が生成される楽音の音高の周期に対応するようになる。物理モデル音源は、管楽器系とされた場合に、マウスピースより右側の管をシミュレートする回路において、4つの乗算器MU4~MU7と2つの加算器AD4,AD5 からなる4乗算格子型ジャンクションは、トーンホールをシミュレートしており、マウスピースからトーンホール間の管の伝搬遅延が遅延回路DELAY-RLによりシミュレートされ、トーンホールから管の終端間の伝搬遅延が遅延回路DELAY-RRによりシミュレートされている。さらに、管の音響損失はローパスフィルタFILTER-Rでシミュレートされ、管終端部での反射は、乗算器MU8によりシミュレートされている。

【0026】

同様に、マウスピースから左側の管をシミュレートする回路において、その管の伝搬遅延は遅延回路DELAY-Lによりシミュレートされており、管の音響損失はローパスフィルタFILTER-Lでシミュレートされ、管終端部での反射は、乗算器MU3によりシミュレートされている。

なお、遅延回路DELAY-RL、DELAY-RR、DELAY-Lには、生成される楽音のピッチ

に応じてテーブルから読み出された遅延時間DRL、DRR、DL がそれぞれ設定され、ローパスフィルタFILTER-R、FILTER-Lには選択された音色を得るためのフィルタパラメータFRP、FRL がそれぞれ供給され、乗算器MU4～MU7 には、トーンホールの開閉操作によって変化する波形伝搬態様をシミュレートすべく、トーンホールの開閉操作に対応した乗算係数M1～M4 が供給される。

【0027】

この場合、遅延回路DELAY-RL、DELAY-RR、DELAY-Lに設定される遅延時間の総和により出力信号のピッチがほぼ決定されることになるが、ローパスフィルタFILTER-R、FILTER-Lには動作遅延時間が生じるため、この動作遅延時間を差し引いた遅延時間を遅延回路DELAY-RL、DELAY-RR、DELAY-Lに振り分けて設定するようにする。

また、マウスピースは右側の管をシミュレートする回路からの反射信号に乗算係数J2を乗算する乗算器MU2と、左側の管をシミュレートする回路からの反射信号に乗算係数J1を乗算する乗算器MU1とを備えている。さらに、乗算器MU1とMU2との出力信号を加算器AD1で加算して、右側の管をシミュレートする回路に出力するとともに、左側の管をシミュレートする回路に出力している。この場合、それぞれ減算器AD2、AD3により、それぞれの管をシミュレートする回路からの反射信号が減算されて、それぞれの管をシミュレートする回路に供給されている。

【0028】

さらに、加算器AD1には係数J3が乗算された励振部からの励振信号EX OUTが供給されて、励振戻り信号EXT INが加算器AD6を介して励振部に戻されている。なお、励振部はマウスピースの一部を構成している。

この物理モデル音源からの出力は、ループのどの部分から出力してもよいが、図示する場合は、遅延回路DELAY-RRからの出力信号が出力信号OUTとして出力されている。出力された出力信号OUTは、図5に示すエンベロープ(ENVELOPE)制御部ELに入力されてエンベロープパラメータ(EG PAR)に基づくエンベロープが付与される。このエンベロープパラメータには、キーオン時のアタックレートやキーオフ時のリリースレートのパラメータが含まれている。

【0029】

さらに、共鳴体モデル部 R E に続いて入力されて、供給された共鳴体パラメータ (RESONATOR PAR) に基づいて楽器胴体の共鳴フォルマントの付与が行われる。さらに、エフェクト部 E F に入力されて、供給されたりバース、コーラス、ディレイ、パン等の各種音響効果を付与する効果パラメータ (EFFECT PAR) に基づいて効果付与が行われて楽音信号 TONE OUT として出力される。なお、ここでいう楽音信号 TONE OUT は、音源波形データと同義であり、所定のサンプリング周期のサンプルデータとされている。

【0030】

図 6 にマウスピースの一部を構成する励振部の構成例を示すが、励振部戻り信号 EX IN は、マウスピース内においてリードに向かって帰還される空気振動波の圧力に相当する信号として減算器 AD11 に供給され、吹奏圧信号 P が減算される。減算器 AD11 からの出力は、マウスピース内の圧力に相当する信号となり、マウスピース内の圧力変化に対するリードの応答特性をシミュレートした励振部フィルタ FIL10、および、マウスピース内の空気流の流速のマウスピース内空気圧に対する飽和特性をシミュレートした非線形変換部 2 (NLC2) に乗算器 MU11 でゲイン調整が行われて入力される。励振部フィルタ FIL10 は、供給されるフィルタパラメータ E F により、カットオフ周波数や選択度が制御されている。

【0031】

励振部フィルタ FIL10 からの出力信号は、乗算器 MU10 においてゲイン調整が行われて、演奏者がマウスピースを咥える圧力に相当するアンプシールド信号 E が加算器 AD10 において加算され、リードに加わる圧力に相当する信号が出力される。この加算器 AD10 からの出力信号は、リードの開閉特性をシミュレートする非線形変換部 1 (NLC1) に供給され、非線形変換部 1 の出力と、非線形変換部 2 の出力とが乗算器 MU12 において乗算され、マウスピースおよびリード間の間隙を通過する空気流の体積流速に相当する信号が乗算器 MU12 から出力される。乗算器 MU12 から出力される信号は、乗算器 MU13 においてゲイン調整が行われて、励振信号 EX OUT として出力される。

【0032】

以上の説明は、管楽器をシミュレートした場合の音源モデルの説明であり、弦



楽器をシミュレートしたときは、マウスピースをシミュレートする回路は、弦に振動が与えられる擦弦部あるいは撥弦部をシミュレートすることになる。すなわち、信号Pが撥弦力や弓速に対応する励起信号となり、信号Eが弓圧に相当する信号となる。なお、弦楽器をシミュレートする場合は、乗算器MU11に供給する乗算係数NL2Gをほぼゼロとすると共に、非線形変換部2の出力を所定の固定値（例えば「1」）として、非線形変換部2（NLC2）の機能を用いていない。

また、遅延回路DELAY-RL、DELAY-RR、DELAY-Lは、弦の伝搬遅延時間をシミュレートするものであり、ローパスフィルタFILTER-R、FILTER-Lは弦の音響伝搬損失をシミュレートするものとなる。

励振部においては、乗算係数NLG1,NLG2,NL1,NL2を種々設定することによって、モデリング対象の楽器に応じた励振部を形成することができる。

【0033】

次に、RAM3上に展開される各種データについて図7を参照しながら説明する。

前記したようにソフトウェア音源モジュールSSMが起動されると、その内部のMIDI出力ドライバ部が起動されて、入力されたMIDIメッセージに応じた各種の音制御パラメータがRAM3に格納される。特に、MIDIメッセージにより前記図4ないし図6に示すような物理モデル音源（VA音源ともいう。）が指定されている場合は、選択されたVA音源用の音制御パラメータVATONEPARがRAM3上に設定されている制御パラメータバッファ（VATONEBUF）に格納される。また、ソフトウェア音源モジュールSSMにより1フレーム毎に生成演算された音源波形データは、RAM3に設定された波形出力バッファ（WAVEBUF）に格納される。さらに、インターフェース”MIDI API”を経由して入力されるMIDIメッセージの内容とそのメッセージの受信時刻（イベントタイム）が対とされて、RAM3上に設定されているMIDI入力バッファ（MIDI RCV BUF, TM）に格納される。さらにまた、RAM3上にはCPU1のワークエリア（CPU working area）が設定されている。

【0034】

バッファVATONEBUFには、選択された図8に示すような音制御パラメータVATON

EPARが格納されると共に、演算生成されるサンプルデータの動作サンプリング周波数を示すパラメータSAMPFREQ、MIDIメッセージ内のイベントがVA音源を指定するキーオンイベントの時にセットされるキーオンフラグVAKEYON、その場合のピッチを指定するパラメータPITCH(VAKC)、および、ベロシティを指定するパラメータVAVEL、ブレスコントローラの操作量パラメータBRETH CONTが格納される。さらに、息圧あるいは弓速度の値が格納されるプレッシャーバッファPBUF、ピッチベンドパラメータが格納されるPBBUF、アンプシュール信号あるいは弓圧信号が格納されるアンプシュールバッファEMBBUF、VA音源における発音トランケートを指定するフラグVAKONTRUNCATE、ボリウムその他のパラメータの格納バッファmiscbuf が設定されている。

【0035】

上記したパラメータSAMPFREQは、たとえば2つのサンプリング周波数のうちの一方を選択可能とされており、第1サンプリング周波数を44.1kHz、第2サンプリング周波数を第1のサンプリング周波数の1/2の22.05kHzとすることができる。また、第2サンプリング周波数を2倍の88.2kHzとしてもよいが、ここで掲げたサンプリング周波数は一例であり、本発明のサンプリング周波数を特定するものではない。

ところで、サンプリング周波数FSを $1/2 \cdot FS$ とすると、1フレームにおいて音源波形サンプルの生成演算数の回数は、FS時に比較して1/2でよいことになる。これを利用して、本発明では、CPU1の負荷状態をチェックして負荷が重いと判定された場合に $1/2 \cdot FS$ とされたサンプリング周波数を選択し、CPU1の負荷を下げることによって演算生成できないサンプルが生じないようにしている。

【0036】

また、サンプリング周波数FSを2FSとすると、1フレームにおいて音源波形サンプルを生成演算する回数は、FS時に比較して2倍となり高精度のデータを生成することができる。そこで、CPU1の負荷状態をチェックして負荷が軽いと判断された場合に、2FSとされたサンプリング周波数を選択して高精度の音源波形データのサンプルを生成演算することができる。

なお、本発明では基準とされるサンプリング周波数を $FS1$ とすると、選択できるサンプリング周波数 $FS2$ は、第1の例では

$$FS1 = n \cdot FS2 \quad (n \text{ は整数})$$

となり、第2の例では

$$FS1 = 1/n \cdot FS2 \quad (n \text{ は整数})$$

となる。

本発明は、第1の例を主体としており、以下の説明では第1の例を主体として説明を行う。

【0037】

なお、本発明は、生成される音源波形データのサンプリング周波数を選択可能としたが、他にもCODECで再生したい音響信号がある場合は、CODECのDAコンバータのサンプリング周波数を特定の標準的な値に固定しておくのがよい。例えば、音楽CDのデジタル出力に本発明による楽音をミキシングする場合などは、CDの規格に合わせて $FS1 = 44.1 \text{ kHz}$ にしておけばよい。

このように、CODECのDAコンバータのサンプリング周波数が標準的な値に固定されると共に、演算生成される音源波形データにおける変更されるサンプリング周波数の関係が $FS1 = n \cdot FS2$ とされた場合（上記第1の例）の本発明の実施の形態の例について次に説明する。

【0038】

DAコンバータのサンプリング周波数が標準的な値に固定されることから、この固定された標準サンプリング周波数 $FS1$ の周期毎に1サンプルずつ読み出される波形出力バッファWAVEBUFには、波形演算のサンプリング周波数の如何に関わらず標準サンプリング周波数 $FS1$ に対応した波形データ列を作成しておく必要がある。そこで、サンプリング周波数 $FS1$ の $1/n$ のサンプリング周波数 $FS2$ が選択された場合は、波形出力バッファWAVEBUFには、演算結果の波形サンプルが隣接したバッファ段に n 個連続して並ぶように書き込んでいく。そして、音源波形データが1フレーム分波形出力バッファWAVEBUFに書き込まれたら、その波形出力バッファWAVEBUFの内容をCODECに引き渡せばよい。

なお、波形出力バッファWAVEBUFのデータ列のサンプリング周波数 FSb と、C

ODEC (DAC) の動作サンプリング周波数 FS_c が異なり、サンプリング周波数のマッチングが必要な場合においては、例えば $FS_b = k \cdot FS_c$ ($k > 1$) である時は、波形出力バッファ WAVEBUF から音源波形データを n アドレスずつ更新しながらの飛ばし読みで CODEC に引き渡せばよい。つまり、波形出力バッファ WAVEBUF に楽音波形サンプルを格納する処理から CODEC の DAC に至るまでの間に、サンプリング周波数変換処理あるいはサンプリング周波数変換回路を挿入して、サンプリング周波数を合わせるようにすればよい。

【0039】

なお、MIDI イベントタイムバッファ TM に格納される時間情報は、イベント対応処理時に必要となるが、フレーム時間を例えば、5 ms あるいは 2.5 ms などと十分短く設定する場合は、フレーム内での各種イベント対応処理の細かいタイミング管理は実質的に不要となるので、この場合は時間情報を特に考慮してイベント対応処理を行わなくてもよい。

ただし、ブレスコントローラ等からの情報は後着優先とするのが好ましいので、このような情報のイベントについては前記時刻情報を用いて後着優先処理を行うようにする。

また、RAM 3 には以上説明したバッファのほかに、アプリケーションプログラムを格納するようにしてもよい。

【0040】

音制御パラメータ VATONEPAR の詳細を示す図表を図 8 に示すが、音制御パラメータ VATONEPAR は、励振部パラメータ (EXCITER PARAMETERS)、管楽器/弦楽器パラメータ (P/S PARAMETERS)、エンベロープパラメータ (EG PAR)、共鳴体パラメータ (RESONATOR PAR)、効果パラメータ (EFFECT PAR)、サンプリング周波数データ (SAMPLING FREQ.) から構成されている。

これらの各パラメータには図 8 に示すような複数のパラメータが含まれているが、各遅延量パラメータや各トーンホールジャンクション乗算係数は音高により決定されるパラメータである。この場合、DL ないし DRR は音高に対する遅延量の遅延量テーブルとされており、総遅延量が所望の音高周期に対応したものなるように各遅延量が遅延量テーブルから読み出されて設定されるようになる。この遅

延量テーブルは、所定の音高の音を実際に発音させて、その音高周波数のずれをフィードバックすることにより作成したテーブルである。

【0041】

また、FLPおよびFRPのフィルタパラメータは、モデリングする管の形状や、弦の特性、および操作子の操作量に応じて設定される。

なお、サンプリング周波数に応じて好適な音制御パラメータVATONEPARが設定されており、この音制御パラメータVATONEPARのサンプリング周波数がSAMPLING FREQ.で示されている。生成演算処理時には、このSAMPLING FREQ.情報を参照して該当するサンプリング周波数用に用意された音制御パラメータVATONEPARを用いて演算処理を行う。この場合、標準サンプリング周波数がFS1とされ、サンプリング周波数FS2は、たとえば $1/2 \cdot FS1$ とされている。

【0042】

次に、フローチャートを参照しながら本発明の動作を詳細に説明する。

図9は電源投入時等に実行される初期プログラムのフローチャートであり、初期プログラムがスタートされると、ステップS10にてハードウェアの初期化等のシステムの初期化が実行される。そして、ステップS11にてOSプログラムが起動されて、他のプログラムを実行できる状態となり、たとえばメインプログラムが実行されるようになる。

【0043】

次に、CPU1により実行されるメインプログラムのフローチャートを図10に示すが、メインプログラムがスタートされると、ステップS20にて各種レジスタをリセットする等の初期設定が実行される。ついで、ステップS21にてデスクトップ等のディスプレイ画面に対するウィンドウの配置等の基本的なディスプレイ処理を行う。その後、ステップS22にてタスク・スイッチングのために起動要因のチェックがおこなわれる。そして、ステップS23にて起動要因が発生されたか否かの判定が行われ、起動要因が発生するまではステップS21ないしステップS23の処理が循環して行われる。ここで、起動要因が発生されると、ステップS23にてYESと判定されてステップS24に進み、起動要因に応じたタスクが実行されるようにタスク・スイッチング処理が行われる。

【0044】

タスクのスイッチングを行う要因は要因1ないし要因5まであり、まず、アプリケーションソフト等からソフト音源API ("MIDI API") を介してMIDIメッセージが供給されたことが検出されたときに要因1に該当するものとして、ステップS25にてソフト音源モジュールSSMが起動されてMIDI処理が行われる。また、1フレームの周期ごとに割り込みが出力されるソフトタイマ (tim) からの内部割り込みがあったときには、要因2に該当するものとして、ステップS26にてソフト音源モジュールSSMが起動されて、波形演算処理が実行され所定サンプル数の音源波形データが生成される。さらに、DMA方式とされている出力デバイス (CODEC) から音源波形データの転送要求があったときは、要因3に該当するものとしてステップS27にて、波形出力バッファWAVEBUFに格納されている音源波形データが出力デバイスに転送される転送処理が行われる。

【0045】

さらにまた、処理装置に備えられているマウスやキーボードなどの入力操作子の操作に基づく操作イベントが検出されたときには要因4に該当するものとして、音色設定の操作イベントの場合は、ステップ28にて音色設定処理が行われ、その他の操作イベントの場合はステップS29にてその他の処理が行われる。さらにまた、終了操作を検出したときには要因5に該当するものとして、ステップS30にて終了処理が実行される。

なお、起動要因がないときは要因4に該当するものとしてステップS28およびステップS29の処理を行う。

そして、これらの要因1ないし要因5に対する処理が終了すると、ステップS21に戻り、ステップS21ないしステップS30の処理が循環して、常時行われるようになっている。

【0046】

次に、前記ステップS25にて実行されるMIDI処理のフローチャートを図11に示す。

MIDI処理がスタートされると、ステップS40にて供給されたMIDIイベントの内容のチェックが行われる。このチェックは、具体的にはバッファとして構成さ

れている”MIDI API”に書き込まれているMIDIメッセージに対して行われる。そして、そのMIDIイベントがノートイベントか否かがステップS41にて判定されて、ノートオンイベントと判定されるとステップS42に進み、そのノートオンイベントに割り当てられている発音チャンネル(MIDI CH)が物理モデル音源(VA音源)に割り当てられた発音チャンネル(VA CH)か否かがステップS42にて判定される。

ノートオンイベントに割り当てられている発音チャンネルが物理モデル音源と判定されると、ステップS43に進み物理モデル音源におけるキーオン処理が実行されてリターンされる。また、ノートオンイベントに割り当てられている発音チャンネルが物理モデル音源でないと判定されると、ステップS44に分岐して他の音源のキーオン処理が行われ、リターンされる。このキーオン処理は、たとえば音源ボード(DSP BOARD) 9上のDSP9-1にて行われる。

【0047】

また、ステップS41にてノートオンイベントでないと判定されると、ステップS45に分岐してノートオフイベントか否かが判定される。ここで、ノートオフイベントと判定されるとステップS46に進み、そのノートオフイベントに割り当てられている発音チャンネル(MIDI CH)が物理モデル音源(VA音源)に割り当てられた発音チャンネル(VA CH)か否かが判定される。そして、ノートオフイベントに割り当てられている発音チャンネルが物理モデル音源と判定されると、ステップS47に進み物理モデル音源におけるキーオンフラグVAKEYONが「0」とされると共に、MIDIイベントタイムバッファTMにノートオフイベントの発生時刻が格納されて、リターンされる。

また、ノートオフイベントに割り当てられている発音チャンネルが物理モデル音源でないと判定されると、ステップS48に分岐して他の音源のキーオフ処理が行われリターンされる。

【0048】

さらに、ステップS45にてキーオフイベントでないと判定されると、ステップS49に分岐して、MIDIイベントがプログラムチェンジか否かが判定される。ここで、プログラムチェンジのMIDIイベントと判定されるとステップS50に進

み、そのプログラムチェンジのMIDIイベントに割り当てられている発音チャンネル (MIDI CH) が物理モデル音源 (VA音源) に割り当てられた発音チャンネル (VA CH) か否かが判定される。そして、プログラムチェンジのMIDIイベントに割り当てられている発音チャンネルが物理モデル音源と判定されると、ステップS 5 1に進みプログラムチェンジで指定されているVA音源用の音制御パラメータVATONEPARがバッファVAPARBUFに格納されてリターンされる。

また、プログラムチェンジのMIDIイベントに割り当てられている発音チャンネルが物理モデル音源でないと判定されると、ステップS 5 2に分岐してその発音チャンネルに対応する音源の音色パラメータ処理が行われリターンされる。

【0049】

さらにまた、ステップS 4 9にてプログラムチェンジのMIDIイベントでないと判定されると、ステップS 5 3に分岐して該当するMIDIイベントのMIDIイベント処理が行われてリターンされる。このMIDIイベント処理では、後述するように、たとえばブレスコントローラを操作したとき等の処理が行われる。

【0050】

次に、前記ステップS 4 3にて実行される物理モデル音源キーオン処理のフローチャートを図12 (a) に示す。物理モデル音源キーオン処理がスタートされると、ステップS 5 5にて受信されたMIDIメッセージ中のノートナンバーがパラメータVAKCとしてバッファVATONEBUFに、MIDIメッセージ中のベロシティ情報がパラメータVAVELとしてバッファVATONEBUFに格納され、VAKEYONフラグが「1」とされる。さらに、MIDIメッセージの受信時刻がイベント発生時刻としてバッファATMに、パラメータVAKCとピッチベンド値が格納されているピッチベンドバッファPBBUFに格納されたピッチベンド値から変換したピッチ周波数データがパラメータPITCHとしてバッファVATONEBUFに格納される。以上の処理が終了するとリターンされる。

なお、ピッチ周波数を求めるときにピッチベンド値を使用することに替えて、アンブシュールパラメータの設定にピッチベンド値を使用してもよい。

【0051】

次に、前記した要因4が生じたときにステップS 2 8にて実行される音色設定

処理のフローチャートを図12(b)に示す。ユーザがマウスやキーボードを操作して音色の設定操作を行うと、音色設定処理が起動され、ステップS50にて物理モデル音源の音色設定が指示されたか否かが判定される。ここで、物理モデル音源の音色設定が指示されたと判定されると、ステップS61にて指示された音色に対応する音色パラメータが、前記図7に示すようにバッファVATONEBUF上に展開される。次いで、ステップS62にてユーザによる音色パラメータの編集が行われて、音色設定処理は終了する。また、ステップS60にて物理モデル音源の音色設定が指示されていないと判定された場合は、ステップS62にジャンプしてユーザによる音色パラメータの編集が行われて、音色設定処理は終了する。

【0052】

次に、前記ステップS53にて行われるその他のMIDIイベント処理のフローチャートを図12(c)に示す。

その他のMIDIイベント処理がスタートされると、ステップS65にて受信されたその他のMIDIイベントに割り当てられている発音チャンネル(MIDI CH)が物理モデル音源(VA音源)に割り当てられた発音チャンネル(VA CH)か否かが判定される。そして、その他のMIDIイベントに割り当てられている発音チャンネルが物理モデル音源と判定されると、ステップS66に進みその他のMIDIイベントがブレスコントローライイベントか否かが判定され、ブレスコントローライイベントと判定された場合は、ステップS67にてブレスコントローライイベント内のパラメータBRETH CONTがブレッシャーバッファPBUFに格納される。

【0053】

また、ブレスコントローライイベントと判定されない場合は、ステップS67をジャンプしてステップS68に進み、その他のMIDIイベントがPITCH BENDイベントか否かが判定される。ここで、PITCH BENDイベントと判定されるとステップS69にてアンブシュールモードとされているか否かが判定されて、アンブシュールモードと判定されたときは、ステップS70にてPITCH BENDイベント中のパラメータPITCHBENDがアンブシュールバッファEMBBUFに格納される。また、アンブシュールモードでないと判定された場合は、ピッチベンドとして使用するよう

ステップS 7 2にてPITCH BENDイベント中のパラメータPITCHBENDがピッチベンドバッファPBBUFに格納される。

【0054】

さらに、ステップS 6 5にて物理音源モデルに割り当てられた発音チャンネルでないと判定された場合、および、ステップS 6 8にてPITCH BENDイベントでないと判定された場合は、ステップS 7 1にジャンプして、受信されたその他のMIDIイベントが上記イベントに該当しないとして、ステップS 7 1にてそのイベントに対応した処理が行われてリターンされる。

なお、アンプシュール信号は演奏者がマウスピースを咥える圧力に相当する信号であり、アンプシュール信号によりピッチが変化するため、アンプシュールモード時にはパラメータPITCHBENDがアンプシュールバッファEMBBUFに格納されるのである。

以上説明したように、MIDIイベントが受信されるごとにMIDIイベント処理により演奏に関わるパラメータの書き換えが行われるのである。

【0055】

次に、物理モデルパラメータ展開処理のフローチャートを図13に示すが、この処理は前記した音色設定処理のステップS 6 1にて行われる処理であり、発音させる前に行われる処理である。

物理モデルパラメータ展開処理がスタートされると、ステップS 7 5にてCPU負荷状態がチェックされる。このチェックはCPU1からの状態報告等に基づいて、設定されているサンプリング周波数FSの値を考慮して行われる。そして、チェックの結果、ステップS 7 6にてCPU1に余裕があると判定されたときは、ステップS 7 7にてユーザが設定した1フレームの周期、またはもっとも短いフレーム周期である標準値のフレーム周期TIMDEFが、波形発生処理を起動するタイマ割り込みを出力するソフトタイマの周期timとして設定される。

なお、標準値のフレーム周期TIMDEFは、たとえば2.5msecとされる。

【0056】

ついで、ステップS 7 8にて選択されている物理モデル音源用の音制御パラメータVATONEPAR内のサンプリング周波数FSが、動作サンプリング周波数SAMPFRE

Qとして設定される。さらに、ステップS 7 9にてアラーム解除処理が行われ、ステップS 8 0にてパラメータSAMPFREQ、パラメータVAKCに対応する音制御パラメータVATONEPARが読み出されてバッファVAPARBUFに格納されてリターンされる。この場合、パラメータVAVELをも考慮した音制御パラメータVATONEPARを読み出してバッファVAPARBUFに格納してもよい。

【0057】

また、ステップS 7 6にてCPU 1に余裕がないと判定された場合は、ステップS 8 1に分岐してフレーム時間自動変更モードに設定されているか否かが判定される。フレーム時間自動変更モードに設定されていると判定された場合は、ステップS 8 2にて標準値のフレーム周期TIMDEFに整数 α が乗算された値が、ソフトタイマの周期timとして設定される。整数 α は1を越える任意の設定数であり、フレーム周期を伸ばすと、パラメータを物理モデル音源に対して送る頻度を少なくすることができ、変更データの転送処理やデータ変化に伴う演算などを低減することができる。

【0058】

ついで、ステップS 8 3にて現在設定されている動作サンプリング周波数SAMPFREQのチェックが行われるが、動作サンプリング周波数SAMPFREQが標準のサンプリング周波数FS 1とされている場合は、CPU 1に余裕がないことからステップS 8 4にて $1/n$ とされたサンプリング周波数FS 2が動作サンプリング周波数SAMPFREQとして設定される。そして、前記したステップS 7 9以降の処理が行われる。ただし、ステップS 8 0にては、変更されたパラメータSAMPFREQに対応する新たな音制御パラメータVATONEPARが読み出されてバッファVAPARBUFに格納されるようになる。

【0059】

さらに、ステップS 8 3にて動作サンプリング周波数SAMPFREQが標準サンプリング周波数FS 1とされていないと判定された場合は、ステップS 8 5に分岐してアラーム表示の処理が行われる。これは、動作サンプリング周波数SAMPFREQが $1/n \cdot FS 1$ とされ、CPU 1の負荷が比較的小さくなるはずのサンプリング周波数FS 2に設定されているにも関わらず、CPU 1に余裕がないと判断され

たからであり、この場合は物理モデル音源における演算生成処理が不可能となるからである。ついで、ステップS86にて物理モデル音源が発音中であれば、その消音処理を行い前記したステップS80の処理を行う。

【0060】

以上説明した処理により、バッファVAPARBUFに物理モデル音源が波形データを生成するための音制御パラメータVATONEPARが格納され、演算による波形発生処理が行われるようになる。この波形発生処理においては、CPU1の負荷状態によって動作サンプリング周波数をダイナミックに変更する処理が同時に行われる。この物理モデル音源の波形発生処理のフローチャートを図14および図15に示す。

周期timが設定されたソフトタイマのタイマ割り込み出力により波形発生処理が起動され、ステップS90にてキーオンフラグVAKEYONが「1」とされているか否かが判定される。キーオンフラグVAKEYONが「1」とされていると判定されると、ステップS91にてそのフレームで必要とされる演算量が計算される。この演算量は、継続音のためのサンプル数と、直前のフレームにおいて受信されたMIDIメッセージにキーオンイベントが含まれていた場合は、新たに発音される音のサンプル数とを生成するための演算量となる。なお、新たに発音される音のサンプル数は、MIDIメッセージを受信した時刻から当該フレームが終了するまでに必要なサンプル数でよい。

【0061】

ついで、ステップS92にてCPU1の負荷状態がチェックされる。このチェックは前フレームにおいて波形演算時間が1フレーム周期に対して占めた割合等を考慮して行われる。そして、チェックの結果、ステップS93にてCPU1に余裕があると判定されたときは、ステップS94にて選択されている音制御パラメータVATONEPAR内のサンプリング周波数FSが動作サンプリング周波数SAMPFEQとして設定される。また、CPU1に余裕がないと判定された場合は、ステップS105に分岐して動作サンプリング周波数SAMPFEQを下げるか否かが判定される。ここで、動作サンプリング周波数SAMPFEQを下げる余地があると判定された場合は、ステップS106にて動作サンプリング周波数SAMPFEQが $1/n$ とされ

たサンプリング周波数FS2とされる。

【0062】

また、すでにサンプリング周波数FS2とされていて動作サンプリング周波数SAMPFREQを下げる余地がないと判定された場合は、ステップS107に分岐してアラーム表示の処理が行われる。これは、動作サンプリング周波数SAMPFREQが $1/n \cdot FS1$ とされ、CPU1の負荷が比較的小さくなるはずのサンプリング周波数FS2に設定されているにも関わらず、CPU1に余裕がないと判断されたからであり、この場合は必要とする演算量を1フレーム時間あるいは所定時間内で達成することができないからである。ついで、ステップS108にて物理モデル音源が発音中であれば、その消音処理が行われリターンされる。

【0063】

ところで、ステップS94あるいはステップS106の処理が終了すると、ステップS95にてアラーム解除処理が行われ、ステップS96にて動作サンプリング周波数SAMPFREQの変更処理が行われたか否かが判定される。ここで、動作サンプリング周波数SAMPFREQの変更処理が行われたと判定された場合は、ステップS97にて、サンプリング周波数変更によるパラメータ変更処理が行われる。すなわち、発音中のノートに関する動作サンプリング周波数SAMPFREQに対応する音制御パラメータVATONEPARが読み出されてバッファVAPARBUFに格納される。なお、動作サンプリング周波数SAMPFREQの変更処理が行われないと判定された場合は、ステップS97の処理はスキップされる。

【0064】

ついで、ステップS98にてトランケート処理を行うか否かが判定される。このトランケート処理は、単音仕様の場合のトランケート処理とされ、発音されていた音の消音処理を行って、次の音を立ち上げる処理が行われる。ここで、トランケートフラグVATRUNCATEが「1」とされていると、YESと判定されてトランケート処理が行われる。すなわち、ステップS99に進み息圧あるいは弓速度の信号Pと、アンプシュールあるいは弓圧力の信号Eが「0」とされ、ステップS100にてエンベロープのフォーシングダンプ処理が行われる。このダンプ処理は、ENVLOPE制御部に供給されるEGPARを制御することにより行われる。そして、

エンベロープのフォーシングダンプ処理が終了したか否かがステップS101にて判定され、終了したと判定されると、ステップS102にてループ内の遅延回路に設定されているDELAY量が「0」とされる。これにより、発音されていた音の消音処理が終了する。

【0065】

続けて、図15に示すステップS109にてプレッシャーバッファPBUFに格納されているデータが信号Pとして、アンプシュールバッファEMBBUFに格納されているデータが信号Eとして設定されると共に、キーコードパラメータVAKCとピッチベンドバッファPBBUFに格納されていたピッチベンドパラメータに基づいて変換した周波数データを、ピッチパラメータPITCHとして設定する。さらに、ステップS110にてバッファVAPARBUFに格納されている物理モデル用の音制御パラメータVATONEPARに基づいて、物理モデル演算処理が行われる。この演算処理が一度行われると1サンプルの音源波形データが生成され、1サンプルの音源波形データは波形出力バッファWAVEBUFに格納される。

【0066】

そして、ステップS111にて前記ステップS91にて計算したサンプル数分のサンプルの演算が終了したか否かが判定され、演算が終了していないと判定された場合は、ステップS113に分岐して、例えば1フレーム時間あるいは所定時間内におけるCPU1の演算占有時間がチェックされる。チェックの結果1フレーム時間を超えていないとステップS114にて判定された場合は、ステップS110に戻り次のサンプルの演算処理が行われる。ステップS110、S111、S113、S114の処理は、1フレーム時間を超えない限り所定のサンプル数が得られるまで循環して実行される。この結果、ステップS111にて1フレームにおける所定のサンプル数分の演算が終了したと判定されるようになる。そして、ステップS112にて波形出力バッファWAVEBUFに格納された音源波形データが出力デバイス(CODEC)に引き渡されるようになる。

【0067】

また、所定サンプル数分の演算が終了しないうちに1フレーム時間を超えてしまったとステップS114にて判定された場合は、ステップS115に分岐して

波形出力バッファWAVEBUF上の音源波形データの消音処理が行われる。ついで、ステップS112にて波形出力バッファWAVEBUFに格納された音源波形データが出力デバイス(CODEC)に引き渡されるようになる。

なお、ステップS90にてキーオンフラグVAKEYONが「1」とされていないと判定された場合は、ステップS103に分岐されキーオフ処理中か否かが判定され、YESと判定された場合はステップS104にてキーオフ処理が続行される。また、キーオフ処理中と判定されない場合は、そのままリターンされる。

【0068】

ところで、サンプリング周波数に変更可能な物理モデル音源を構築するためには、サンプリング周波数が可変できると共に、遅延時間も自由に設定できる遅延手段が必要になる。この可変遅延手段について図18を参照しながら次に説明する。

【0069】

物理モデル音源においては、各遅延回路はRAM3の遅延領域をシフトレジスタとして使用することにより所定の遅延量を得るようにしている。図18に示す $DELAY \times 20$ がRAM3上に確保された遅延領域からなる遅延回路であり、その遅延量の整数部は入力されるデータが書き込まれるアドレス位置を示すライトポインタ(WRITE POINTER)と、データが読み出されるアドレス位置を示すリードポインタ(READ POINTER)との間のシフトレジスタ段数Dとなる。また、遅延量的小数部は、リードポインタと該リードポインタよりn段前のアドレス位置(READ POINTER-n)からそれぞれ読み出されるデータ間を補間するための乗算器MU21に設定される乗算係数dとなる。ただし、補間用の乗算器MU20には乗算係数(1-d)が設定される。

【0070】

この場合、 $DELAY \times 20$ において加算器AD20から出力される遅延出力の総遅延量は遅延段数換算では、 $(D+d)$ となり、時間換算ではサンプリング周波数FSに対して $(D+d)/FS$ となる。ここで、設定選択可能なサンプリング周波数のうち最高値をFS1とすると、基本的にサンプリング周波数FS1の周期時間が遅延1段に対応するように遅延を構成することが望ましいことになる。そうす

れば、サンプリング周波数を $FS1$ の n 分の 1 にするときは、サンプル演算毎に、隣接した n 段の遅延に演算結果の 1 サンプルを n アドレスにわたり連続書き込みし、かつ遅延出力はリードポインタを n アドレスずつ更新させて読み出せばよいことになる。従って、このような構成の場合、必要な遅延時間 T_d を実現する遅延段数換算値 $(D+d)$ は、サンプリング周波数に関わらず、 $(D+d) = T_d \cdot FS1$ となる。

なお、ライトポインタおよびリードポインタは、等価的にシフトレジスタ上を矢印で図示するアドレス方向に移動するようにされ、シフトレジスタの右端に達するとその左端にジャンプするように移動して、 $DELAY \times 20$ 上を循環している。

【0071】

上記したように、サンプリング周波数 FS に関わらず遅延 1 段の時間換算された遅延時間長を一定 ($1/FS1$) としているので、サンプリング周波数 FS が $FS1$ の $1/n$ のサンプリング周波数 $FS2$ に変更された場合に遅延出力の遅延時間長が変化しないように、ライトポインタは演算生成された 1 サンプルデータを連続する n アドレスにわたって書き込みようにして、1 サンプルデータが演算生成される毎に n アドレス進むようにし、リードポインタは n アドレスずつ更新されて $(n-1)$ アドレス飛ばしで遅延されたサンプルデータを読み出すようにしている。これにより、演算生成されたサンプルデータの 1 サンプル前の遅延出力は、 n アドレス前のアドレス位置から読み出した遅延出力に対応するようになる。

このため、図 18 に図示するように小数遅延部においてリードポインタから n 段 (n アドレス) 前のアドレス位置から補間のための 1 サンプル前のデータを読み出しているのである。

【0072】

また、物理モデル音源におけるフィルタ等に備えられる単位遅延手段においても、前記した遅延回路と同様の手段を用いることにより、設定されたサンプリング周波数に変更されても遅延時間長が変更されないようにしている。この単位遅延手段について図 19 を参照して説明する。

単位遅延手段も RAM 3 の遅延領域をシフトレジスタとして使用することによ

り単位遅延を得るようにしている。図19示すDELAYx21がRAM3上に確保された遅延領域からなる単位遅延手段であり、その単位遅延量は入力されるデータを書き込むライトポインタ (WRITE POINTER) のアドレス位置と、データを読み出すリードポインタ (READ POINTER) のアドレス位置との間のn段のシフトレジスタにより得ている。

【0073】

前記図18に示す遅延回路において説明したように、1サンプル前のサンプルデータはnアドレス (n段) 前に書き込まれているので、ライトポインタとリードポインタとのアドレス差はnアドレスとされている。この場合も、ライトポインタはn個同じデータをnアドレスにわたり書き込みようにし、リードポインタはnアドレスずつ更新されてデータを読み出すようにしている。

また、単位遅延手段においては、その性質上n段の遅延領域のみにより構成することが可能である。

【0074】

また、出力デバイスであるCODEC14の再生サンプリング周波数は、前記したように一般に固定されているので演算生成されるサンプルデータのサンプリング周波数が $1/n$ に変更された場合は、音源波形データが1サンプル生成されたときに、その演算生成された1サンプルデータをn個ずつRAM3上の波形出力バッファWAVEBUFの連続するアドレス位置に繰り返し書き込むようにする。従って、この実施の形態の例では波形出力バッファWAVEBUF上には1フレーム分の波形出力サンプルデータ列がサンプリング周波数FS1に対応した形で生成される。ここで、CODEC14がサンプリング周波数FS1で動作するとすれば、CODEC14は波形出力バッファWAVEBUFの内容をそのまま受け取ってサンプリング周波数FS1でDA変換すればよい。

【0075】

なお、CODEC14の再生サンプリング周波数が生成演算されるサンプルデータのサンプリング周波数に応じて同期可変される場合は、演算生成されるサンプルデータを順次1サンプルずつRAM3上の波形出力バッファWAVEBUFに書き込むようにすればよい。

ところで、前記図14および図15に示す波形発生処理においては、サンプリング周波数 FS に最適な音制御パラメータ $VATONEPAR$ を読み出して、音源波形データの生成演算時に使用するパラメータとしてバッファ $VAPARBUF$ に格納している。

このため、各音色の音制御パラメータ $VATONEPAR$ は、取り得るサンプリング周波数 FS 毎のパラメータセットとして記憶手段に記憶されている。

【0076】

この態様の一例を図20(a)に示すが、この例においては、 $VATONEPAR1(FS1)$ と $VATONEPAR1(FS2)$ とは、たとえばピアノの音制御パラメータであり、 $VATONEPARK(FS1)$ と $VATONEPARK(FS2)$ とは、たとえばバイオリンの音色用の音制御パラメータである。このように、 $VATONEPAR1 \sim VATONEPARK$ までの音色番号の音制御パラメータはサンプリング周波数毎に用意された同じ音色のパラメータセットとされている。

また、音色番号が $VATONEPAR(K+1)$ 以降の音制御パラメータは、それぞれ独立した音色とされ、サンプリング周波数 $FS1$ あるいはサンプリング周波数 $FS2$ のいずれかに対応した音制御パラメータとされており、等しくされた音色番号はない。

【0077】

また、他のパラメータの態様の例を図20(b)に示すが、この例においては、設定可能な各サンプリング周波数 FS に対する音色データをそれぞれ同一の音制御パラメータ $VATONEPARi$ に用意するようにしている。すなわち、 $VATONEPAR1(FS1, FS2) \sim VATONEPARm(FS1, FS2)$ までは、サンプリング周波数 $FS1$ 、 $FS2$ 毎に用意された同一の音色のパラメータがすべて1つの音制御パラメータ $VATONEPARi$ 内に用意されている。

この場合、サンプリング周波数 FS に対応する音色パラメータを1つの音制御パラメータ $VATONEPARi$ 内から抽出して、バッファ $VAPARBUF$ に格納するようにする。

【0078】

なお、 $VATONEPARm+1$ 以降の音色番号の音制御パラメータにおいては、独立した

音色の音制御パラメータとされており、サンプリング周波数 $FS1$ あるいはサンプリング周波数 $FS2$ のいずれかに対応する音色用とされている。すなわち、 $VATONEPAR_{m+1}(FS1,*)$ はサンプリング周波数 $FS1$ のパラメータのみとされており、 $VATONEPAR_p(*,FS2)$ はサンプリング周波数 $FS2$ のみのパラメータとされている。

ところで、サンプリング周波数を変更しても聴感上、同じ音になるようにする場合、サンプリング周波数に応じて調整が必要になるパラメータには、遅延ループ部の遅延パラメータをはじめ、他にフィルタ係数や、励振部の非線形変換部の非線形特性等がある。

【0079】

次に、前記した波形発生処理のステップ $S110$ にて実行される物理モデル音源処理のフローチャートを図16に示す。

物理モデル音源処理がスタートされると、ステップ $S120$ にて指定されたピッチ周波数と、各部の設定状態を示す動作サンプリング周波数 $SAMPFREQ$ 、バッファ $VAPARBUF$ に格納されている音制御パラメータ $VATONEPAR$ に応じて、各可変遅延の遅延長設定処理が行われる。この場合の各遅延時間長の設定は、図18に示すように行われる。

【0080】

ついで、ステップ $S121$ にて動作サンプリング周波数 $SAMPFREQ$ 、息圧あるいは弓速度の信号 P 、アンプシュールあるいは弓圧力の信号 E 、および、バッファ $VAPARBUF$ に格納されている音制御パラメータ $VATONEPAR$ に基づいて、前記図6に示すような励振部に関する演算処理が行われる。すなわち、励振戻り信号 $EXIN$ が取り込まれ、ついで動作サンプリング周波数 $SAMPFREQ$ に対応するフィルタパラメータ $FLTPAR$ に基づいて、励振部フィルタ $FIL10$ のフィルタ演算が行われる。さらに、動作サンプリング周波数 $SAMPFREQ$ に応じた非線形変換特性による非線形変換部1、および、必要ならば非線形変換部2の演算と、その周辺部の演算が行われる。そして、励振部出力信号 $EXOUT$ が演算生成されて出力される。

【0081】

続いて、ステップ $S122$ にて動作サンプリング周波数 $SAMPFREQ$ と、バッファ

VAPARBUFに格納されているパラメータVATONEPARに基づいて、前記図4に示すような管／弦モデル部に関する演算処理が行われる。すなわち、励振部出力信号EXOUTが取り込まれ、動作サンプリング周波数SAMPFREQに対応するジャンクションパラメータJUNCTPARに基づいてジャンクション部の演算が行われる。さらに、遅延ループ部の演算処理が行われるが、ここでは、動作サンプリング周波数SAMPFREQに対応するフィルタパラメータFLTPARに基づいて各終端フィルタFILTER-R, FILTER-Lのフィルタ演算も行われる。そして、演算生成された励振戻り信号EXIN、および、出力サンプル信号OUTが出力される。

【0082】

続いて、ステップS123にて動作サンプリング周波数SAMPFREQと、バッファVAPARBUFに格納されているパラメータVATONEPARに基づいて、前記図5に示すような音色効果付与部に関する演算処理が行われる。すなわち、出力サンプル信号OUTが取り出され、ENVELOPE制御部ELの演算処理、共鳴体モデル部REの演算処理、エフェクト部EFの演算処理が行われる。そして、演算生成された最終出力が、音源波形データTONEOUTとして出力される。この音源波形データTONEOUTは前述したように演算生成されたサンプルのサンプリング周波数FSに応じた態様で波形出力バッファWAVEBUFに書き込まれる。

【0083】

次に、物理モデル部演算処理のステップS122にて行われる遅延ループ部演算処理のフローチャートを図17に示す。ただし、このフローチャートでは終端フィルタFILTER-Rと乗算器MU8に関する演算処理だけを詳細に示しているが、FILTER-Lと乗算器MU3の演算処理も同様に行われる。

遅延ループ部演算処理がスタートされると、ステップS130にて終端フィルタFILTER-Rの直前である右終端までのループ部分の演算処理が行われる。ついで、ステップS131にて演算省略条件のチェックが行われる。このチェックは実質的にループゲインがほぼゼロとされている部分の演算を省略して、遅延ループ部全体の演算量を低減するために行う。具体的に演算省略条件を挙げると、第1の演算省略条件は、終端フィルタFILTER-Rの出力が「0」とされている時である。この場合、終端フィルタFILTER-Rから所定時間「0」が連続して出力されると

きとしてもよい。さらに、終端フィルタFILTER-Rの入力や内部遅延レジスタの内容などまでチェックするようにしてもよい。さらにまた、最終出力TONEOUTが十分に減衰している場合も演算省略条件を満足しているとしてもよい。

【0084】

第2の演算省略条件は、終端フィルタFILTER-Rの入力信号に十分な変化がないときである。このときは、演算を省略して直前の終端フィルタFILTER-Rからの出力値を今回の出力値とする。さらに、乗算器MU8においても同様に直前の出力値を今回の出力値としてもよい。

第3の演算省略条件は、乗算器MU8の乗算係数TERMGRがゼロあるいはほぼゼロであるときである。この場合は、演算を省略して右終端出力をゼロとする。

【0085】

以上挙げた演算省略条件の内の終端フィルタFILTER-Rに関する演算省略条件を満足したときに、ステップS132にてYESと判定されてステップS133に進み、満足する省略条件に対応した上述した出力値の受け渡し処理が行われる。また、終端フィルタFILTER-Rに関する演算省略条件を満足しないと判定された場合はステップS137に分岐して終端フィルタFILTER-Rに関する演算が行われる。そして、ステップS133あるいはステップS137の処理が終了すると、ステップS134にて上記の演算省略条件の内の乗算係数TERMGRに関する演算省略条件を満足したときに、YESと判定されてステップS135に進み、満足する省略条件に対応した出力値の受け渡し処理が行われる。また、乗算係数TERMGRに関する演算省略条件を満足しないと判定された場合はステップS138に分岐して、乗算器MU8において乗算係数TERMGRを乗算する演算が行われる。そして、ステップS135あるいはステップS138の処理が終了すると、ステップS136に残る他の遅延ループ部分の演算処理が行われてリターンされる。

【0086】

なお、演算の省略は遅延ループ部分に限らず、励振部、あるいは、音色効果付与部の演算についても省略することができる。励振部については、励振部各部の信号路の信号振幅や関連パラメータの値などがほぼゼロとなっているか等をチェックして演算省略の解析判断をする。

また、音色効果付与部については、ENVELOPE制御部EL、共鳴体モデル部RE、あるいは、エフェクト部EFからの出力が十分減衰されて、ほぼゼロとされているときに、出力がほぼゼロとなっているブロック毎の演算を省略してその出力値をゼロとすればよい。

【0087】

以上説明した本発明において、サンプリング周波数 F_S の変更制御に伴い、非線形部における非線形変換特性によっては折り返しノイズが生じるおそれがある。そこで、このような場合は、非線形変換の入力側においてオーバサンプリングし、得られた非線形変換出力をフィルタで帯域制限した上で元のサンプリング周波数に戻すようにすればよい。

【0088】

ところで、前記図4に示す物理モデル音源において、キーオン中に新たなキーオンが発生したときは、新たなキーオンに対応する楽音を発音する処理を行うようになる。この際、前のキーオンに対応する楽音を継承するような形で発音する場合は、物理モデル内を循環する信号、例えば、管／弦モデル部などの遅延内の信号は、基本的にはそのまま励振信号の発生をキーオンに対応して行えばよい。しかしながら、直前のキーオン対応の楽音発生状態を継承せず、独立性の高い楽音の立ち上げを行う場合、あるいは新たなキーオンで直前のキーオン時とは異なる音色の楽音を発生するときなどは、新たなキーオンに応じて物理モデル内の、特に遅延回路の初期化を行う必要が生じる。

【0089】

この場合に、物理モデル音源の発音チャンネル数が「1」とされている場合には、前記物理モデル音源内の全ての遅延回路を構成しているRAM3の遅延領域をクリアして初期化してから新たなキーオンに対応する楽音を生成することになる。また、物理モデル音源の発音チャンネル数が複数チャンネルとされている場合には、最も減衰の進んでいる発音チャンネルの楽音を消音するように、その発音チャンネルの遅延回路を構成しているRAM3の遅延領域をクリアした後、初期化された当該遅延領域を使用して新たなキーオンに対応する楽音を生成することになる。

【0090】

このようなRAM3の遅延領域のクリアは、当該遅延領域にデータ”0”を書き込むことにより行われるので、クリア時間分だけ楽音生成が遅れるようになる。そこで、遅延領域のクリアの待ち時間をなくすことのできる遅延回路の構成をハードウェア的に図21に示す。

図21に示すように遅延回路は2系列の遅延手段により構成するようにし、第1遅延系列の遅延手段は、乗算手段MU31、遅延手段DELAYaと乗算手段MU32とが縦続接続されて構成されており、第2遅延系列の遅延手段は、乗算手段MU33、遅延手段DELAYbと乗算手段MU34とが縦続接続されて構成されている。そして、入力データINPUTは第1遅延系列および第2遅延系列に分配されて入力され、第1遅延系列および第2遅延系列の出力は、加算手段AD31により合成されて遅延出力データOUTPUTとして出力される。

【0091】

また、乗算手段MU31には乗算係数INGAINaが、乗算手段MU33には乗算係数INGAINbが、乗算手段MU32には乗算係数OUTGAINaが、乗算手段MU34には乗算係数OUTGAINbが、それぞれ与えられている。なお、図21に示すように乗算手段MU31,MU32により入力制御部（INPUT CONTROLLER）が構成されており、乗算手段MU32,MU34および加算手段AD31により合成部（MIX）が構成されている。

ところで、図21においては遅延回路はハードウェア的に表現されているが、実際にはRAM3の遅延領域を用いた遅延処理プログラムによりソフトウェアで実現されている。

【0092】

次に、図21に示す遅延回路の動作を図22を参照しながら説明する。

図22（a）は第1の遅延系列と第2の遅延系列とのいずれかをセレクト的に制御する場合の等価回路を示す図であり、入力データINPUTはセクタ（SEL）31において、遅延手段DELAYaあるいは遅延手段DELAYbのいずれかに導かれる。すなわち、前記した入力制御部がセクタ31を構成することになり、この際に乗算手段MU31に与えられる乗算係数INGAINaと、乗算手段MU33に与えられる乗算係数INGAINbのうち一方の乗算係数を”0”に、他方の乗算係数を”1”に制

御することによりセクタ31の機能を実現している。

【0093】

また、遅延出力データOUTPUTは遅延手段DELAYaあるいは遅延手段DELAYbのいずれかから出力される。すなわち、前記した合成部がセクタ32を構成することになり、この際に乗算手段MU32に与えられる乗算係数OUTGAINaと、乗算手段MU34に与えられる乗算係数OUTGAINbのうち一方の乗算係数が”0”に、他方の乗算係数が”1”に制御されることによりセクタ32の機能が実現されている。

なお、乗算係数INGAINYaと乗算係数OUTGAINaとは等しい係数値となるように制御され、乗算係数INGAINYbと乗算係数OUTGAINbとは等しい係数値となるように制御される。また、遅延手段DELAYaと、遅延手段DELAYbには、それぞれ割り当てられた楽音のピッチに応じた遅延量DLYa、あるいは遅延量DLYbが設定される。

【0094】

図22(a)に示す遅延回路の動作を具体的に説明すると、乗算係数INPUTaと、乗算係数OUTPUTaとが”1”とされ、乗算係数INPUTbと、乗算係数OUTPUTbとが”0”とされていると、入力データINPUTはセクタ31により遅延手段DELAYaに導かれ、遅延手段DELAYaにより設定された遅延量DLYaに応じた時間だけ遅延が行われる。遅延された入力データは、セクタ32を介して所定時間遅延された出力データOUTPUTとして出力される。

また、乗算係数INPUTaと、乗算係数OUTPUTaとが”0”とされ、乗算係数INPUTbと、乗算係数OUTPUTbとが”1”とされていると、入力データINPUTはセクタ31により遅延手段DELAYbに導かれ、遅延手段DELAYbにより設定された遅延量DLYbに応じた時間だけ遅延が行われる。遅延された入力データは、セクタ32を介して所定時間遅延された出力データOUTPUTとして出力される。

【0095】

このような第1の遅延系列と第2の遅延系列とは、トグル的に切り換えて使用することができる。従って、新たなキーオンが発生した時に、例えば、第1の遅延系列を使用していた場合には、新たなキーオンが発生したときに第1の遅延系列における乗算係数INPUTaと乗算係数OUTPUTaとの乗算係数を”1”から”0”

に変更すると同時に、第2の遅延系列における乗算係数INPUTbと乗算係数OUTPUTbとの乗算係数を”0”から”1”に変更するようにする。これにより、第2の遅延系列の遅延手段DELAYbを使用することができ、これにより直ちに新たなキーオンに応じた楽音を生成することができるようになる。この際に、第1系列における乗算係数が”0”に変更されることから、第1の遅延系列の遅延手段DELAYaには、楽音の1周期以内においてデータ”0”が書き込まれるようになり、クリアされるようになる。

【0096】

なお、図22(a)に示す遅延回路はハードウェア的に示されているが、ソフトウェアにより上記した遅延制御を行う場合には、入力側および出力側においてセレクタ31、セレクタ32を設ける必要はなく、キーオン毎にRAM3における空いている遅延領域を割り当てることにより、同等の動作を行うことができる。

また、新たなキーオンにより”0”の乗算係数が設定された遅延系列の遅延手段は、書込ポインタがそれまで使用していた遅延長分（あるいは該当遅延に割り当てられているメモリ領域分）移動することにより”0”が書き込まれてクリアされるが、当該メモリ領域は、次に新たに発生するキーオンにより割り当てられるまで待機状態にしておいてもよい。この際に、メモリ領域に空きフラグを立てるようにするのが好適となる。

さらに、新たなキーオンが発生したときに、トランケート処理が行われて割り当てがはずれた遅延系列のクリア処理を、CPUに余裕があるときに実行するようにしてもよい。

【0097】

次に、図22(b)に示す遅延回路は、図22(a)に示す遅延回路のセレクタ32を合成部(MIX)34に置き換えたものであり、上記した図22(a)に示す遅延回路と同様の遅延制御を行うことができる。なお、図22(b)に示す遅延回路では、セレクタ33により遅延系列を切り換えると同時に、合成部34を構成する乗算器MU32,MU34にそれぞれ設定する乗算係数OUTGAINa,OUTGAINbを徐々に”1”から”0”、あるいは”0”から”1”へ切り換えていくクロスフェ

ード制御を行うこともでき、楽音の1周期以内において、第1の楽音から第2の楽音へ滑らかに移行させることができる。

【0098】

また、図21に示す遅延回路において、乗算係数 $INGAINa$ 、 $INGAINb$ を共に”1”として、第1遅延系列および第2遅延系列を常時並列に動作させておき、キーオンが発生する毎に前回のキーオン時に割り当てられた遅延系列とは異なる側の遅延系列に新たなキーオンに対応したピッチとするための遅延量 DLY を設定するようにする。例えば、前回のキーオン時において第1遅延系列が割り当てられた時には、今回のキーオンでは第2遅延系列の遅延手段 $DELAYb$ に押鍵ピッチに対応する遅延量 $DLYb$ を設定する。同時に、第1遅延系列の乗算係数 $OUTGAINa$ を”1”から”0”に徐々に変化させると共に、第2遅延系列の乗算係数 $OUTGAINb$ を”0”から”1”に徐々に変化させる。

【0099】

このように第1遅延系列と第2遅延系列をクロスフェード制御することにより、実質的に加算手段AD31から出力される出力データ $OUTPUT$ の遅延量は、遅延量 $DLYa$ から遅延量 $DLYb$ に滑らかに変化していくようになる。すなわち、ポルタメントを実現することができるようになる。

さらにまた、第1遅延系列と第2遅延系列とのクロスフェード制御を交互に何度も行おうようにすると共に、クロスフェード制御を行う毎に、徐々に”1”の乗算係数に変化する側の遅延系列の遅延手段に設定される遅延量 DLY を任意に変更することにより、任意のピッチカーブで音高が変化する楽音を得ることができる。

【0100】

さらに、第1遅延系列と第2遅延系列を互いに異なるサンプリング周波数対応の遅延回路として使用すると共に、その遅延量は同一となるようにする。さらに、乗算係数 $INGAINa$ と乗算係数 $INGAINb$ との和が”1”となるようにすると共に、乗算係数 $OUTGAINa$ と乗算係数 $OUTGAINb$ との和が”1”となるようにしながら、適宜各乗算係数を制御する。このようにすると、サンプリング周波数の違いによる音色が混じり合うようになり、新たな音色の楽音を生成することができるようになる。

なる。

また、物理モデル音源における枝路の信号振幅が小さくなった場合には、前記した前のキーオンから新たなキーオンに移行させる方法により、サンプリング周波数の低い遅延系列に移行させるようにしてもよい。この際に移行が完了した時点において、割り当てが解除された遅延系列は他の遅延回路に割り当てることができるようになる。

【0101】

上記した遅延回路においては、RAM3に設定された遅延領域を使用して遅延回路がソフトウェア的に実現されている。この様子を概略的に図23に示す。図23に示すように、RAM3の所定領域が遅延領域（DELAY AREA）に割り当てられており、この遅延領域が複数に分割されて各遅延手段を構成するための単位遅延領域（DELAY1a, DELAY1b, . . . , DELAYa9, . . . , DELAYn）とされる。この単位遅延領域が、それぞれ遅延手段（DELAY1, . . . , DELAYn）に割り当てられるようになる。また、図示するように分割された単位遅延領域毎にフラグ領域を設けるようにして、このフラグ領域に、その単位遅延領域が遅延手段として使用されておらず空いていることを示す「空きフラグ」を立てるようにしてもよい。

【0102】

前記図21に示す遅延回路を実現する遅延領域の割り当てを図23を参照しながら説明する。ただし、物理モデル音源は第1遅延回路ないし第n遅延回路を有しているものとする。

前のキーオンにより、例えば単位遅延領域DELAY1aが第1遅延回路DELAY1の第1遅延系列の遅延手段に割り当てられ、単位遅延領域DELAY1aの遅延量が前のキーオンに係るピッチに応じた遅延量DLYiに設定されている。また、前のキーオンにより、例えば単位遅延領域DELAY9が第n遅延回路DELAYnの第1遅延系列の遅延手段に割り当てられ、単位遅延領域DELAY9の遅延量が前のキーオンに係るピッチに応じた遅延量DLYiに設定されている。

【0103】

次いで、今回のキーオンが発生すると、例えば単位遅延領域DELAY1bが第1遅延回路DELAY1の第2遅延系列の遅延手段に割り当てられ、単位遅延領域DELAY1a

の遅延量が今回のキーオンに係るピッチに応じた遅延量DLYkに設定される。また、今回のキーオンにより、例えば単位遅延領域DELAYnが第n遅延回路の第2遅延系列の遅延手段DELAYnに割り当てられ、単位遅延領域DELAYnの遅延量が今回のキーオンに係るピッチに応じた遅延量DLYkに設定される。

これにより、前記した図21に示す遅延回路の動作を実行することができる。

【0104】

図23に示す構成は、物理モデル音源の発音チャンネルが1発音チャンネルとされている場合を示したが、発音チャンネルが複数発音チャンネルとされている場合に遅延回路を実現する遅延領域の割り当てを図24に示す。

この場合の動作を説明すると、前のキーオンにより、例えば単位遅延領域DELAY1aが第1発音チャンネルの遅延回路DELAY1の第1遅延系列の遅延手段に割り当てられると、単位遅延領域DELAY1aの遅延量が第1発音チャンネルに割り当てられた前のキーオンに係るピッチに応じた遅延量DLYpに設定される。

次いで、今回のキーオンが発生して、例えば単位遅延領域DELAY1bが第1発音チャンネルの遅延回路DELAY1の第2遅延系列の遅延手段に割り当てられると、単位遅延領域DELAY1aの遅延量が第1発音チャンネルに割り当てられた今回のキーオンに係るピッチに応じた遅延量DLYqに設定される。

【0105】

また、前のキーオンにより、例えば単位遅延領域DELAY9が第2発音チャンネルの遅延回路DELAYnの第1遅延系列の遅延手段に割り当てられた場合は、単位遅延領域DELAY9の遅延量が第2発音チャンネルに割り当てられた前のキーオンに係るピッチに応じた遅延量DLYpに設定される。

そして、今回のキーオンにより、例えば単位遅延領域DELAYnが第2発音チャンネルの遅延回路の第2遅延系列の遅延手段DELAYnに割り当てられた場合は、単位遅延領域DELAYnの遅延量が第2発音チャンネルに割り当てられた今回のキーオンに係るピッチに応じた遅延量DLYqに設定される。

これにより、物理モデル音源の発音チャンネルが複音化されていても、前記した図21に示す遅延回路の動作を実行することができる。

【0106】

また、図23および図24に示す構成において、各遅延回路に割り当てられる単位遅延領域は、予め割り当てを固定しておいてもよいが、遅延領域に設定されている単位遅延領域の「空きフラグ」を見ながらキーオン毎にダイナミックに単位遅延領域の割り当てを行うようにしてもよい。

【0107】

以上の説明は主に本発明をパーソナルコンピュータで実施するソフトウェア音源の場合を例にして説明したが、コンピュータシステムの中で、アプリケーション・ソフトとするか、あるいは例えばデバイスドライバ・ソフトにするかのいずれの位置づけで本発明の音源ソフトウェアを起動、動作させるかは、システム構成やオペレーションシステムOSに応じて適宜決めるようにすればよい。

【0108】

さらに、本発明のような音源ソフトウェアまたはその機能を、他のソフトウェア、例えば、ゲームなどのアミューズメントソフト、カラオケソフトや自動演奏・伴奏ソフトなどに組み込んでもよい。さらにまた、オペレーションシステムOSに直接組み込むようにしてもよい。

なお、本発明によるソフトウェアは、フロッピーディスクや光磁気ディスク、CD-ROMなどのディスクメディア、あるいはメモリカードなどで供給することもできる。また、ソフトウェアデータを記憶した半導体チップ（典型的にはROM）を差し替え、追加するようにしてもよい。さらにはNETWORK I/F 11を通じて音源ソフトウェアを配信するようにしてもよい。

【0109】

以上の説明では、パソコンでの応用を例にして説明したが、ゲームやカラオケなどのアミューズメント機器、電子機器、一般電気機器への適用も容易に行うことができる。また、パソコンのオプションとしての音源ボードや音源ユニットに適用することも可能である。さらに、専用MPU（DSP）を使用したソフトウェア処理による音源専用機に適用することもできる。この場合、MPUの処理能力が高ければサンプリング周波数をより高めることができ、高精度の波形出力が必要なときにサンプリング周波数を n 倍にすることもできる。

【0110】

さらに、単音の音源に替えて発音チャンネル数を複音化した場合には、発音処理中のチャンネル数に応じて、サンプリング周波数の可変制御や、演算アルゴリズムの中の省略可能な演算部分の演算削除制御などを行うようにしてもよい。この場合、演奏パート（MIDIチャンネル）毎にサンプリング周波数を設定するようにしてもよい。

さらにまた、以上の実施の形態の説明では、CODECの動作サンプリング周波数は固定であるとしたが、可変であってもよい。その場合、例えば、波形出力バッファWAVEBUFとCODEC（DAC）間にサンプリング周波数を合わせるような処理あるいは回路（典型的にはオーバーサンプリングやダウンサンプリング、データ補間など）を介在させればよい。

【0111】

さらにまた、音色の変化や劣化をさけるために、特定の音色あるいは発音チャンネルの音の生成演算においては、サンプリング周波数の変更処理や演算アルゴリズムの部分的演算削除を禁止するようにしてもよい。

さらにまた、出力ボリュームがゼロに設定された場合は、楽音出力をゼロに下げて、ボリュームが上げられるまでは波形演算処理を実行しないようにしてもよい。

なお、図4ないし図6に示す音源モデルは本発明における音源モデルの一例であり、本発明はこれに限られるものではない。また、本発明で開示されているサンプリング周波数可変手段や、信号レベルなどの所定の条件による波形演算省略制御手段は、物理モデル音源によらず、他の音源方式にも適用することができる。本発明では、特に物理モデル音源への適用を示したが、波形メモリ読み出し方式によるものや、AM、FMなど各種変調技術による音源においても適用可能である。

また、図18および図19に示す遅延時間長の制御方法は、リバース等の信号遅延を用いる各種信号処理にも適用可能である。

【0112】

【発明の効果】

本発明は以上のように構成されているので、CPUの負荷が大きいときはサンプリング周波数を下げるようにすることができ、楽音の発音が途切れることのない

い音源波形データを生成することができる。また、CPUの負荷が軽いときは通常のサンプリング周波数より高いサンプリング周波数とすることができ、高精度の音源波形データを生成することができるようになる。

この場合、サンプリング周波数を変更することに代えて、発音数を変更するようにしてもよい。

【0113】

また、特定の条件を満足する場合、対応の演算を省略するようにしたので、CPUの負荷が極力大きくならないように効率的な演算を行うことができる。したがって、楽音の発音が途切れることのない音源波形データを生成することができる。また、効率的な演算を行えることからサンプリング周波数を従来より高いサンプリング周波数とすることもでき、高精度の音源波形データを生成することができるようになる。

【図面の簡単な説明】

【図1】 本発明の音源波形データ生成方法が実行されるソフトウェアシステムの構成例を示す図である。

【図2】 図1に示すソフトウェアシステムの動作タイムチャートを示す図である。

【図3】 本発明の音源波形データ生成方法を具現化した本発明の音源波形データ生成装置を備える処理装置の一構成例の概略を示すブロック図である。

【図4】 本発明の音源波形データ生成方法における音源モデルの概念を具体化した管／弦モデル部の構成例を示すブロック図である。

【図5】 本発明の音源波形データ生成方法における音源モデルの概念を具体化した音色効果付与部の構成例を示すブロック図である。

【図6】 本発明の音源波形データ生成方法における音源モデルの概念を具体化した励振部の構成例を示すブロック図である。

【図7】 図6に示すRAM3上に展開される各種データを示す図表である。

【図8】 本発明における演算生成に必要な音制御パラメータVATONEPARの詳細を示す図表である。

【図9】 本発明における初期プログラムのフローチャートを示す図である。

【図10】 本発明におけるメインプログラムのフローチャートを示す図である。

【図11】 メインプログラム中のMIDI処理のフローチャートを示す図である。

【図12】 MIDI処理中の物理モデル音源キーオン処理のフローチャート、その他のMIDIイベント処理のフローチャート、および、ユーザにより起動される音色設定処理のフローチャートを示す図である。

【図13】 音色設定処理中の物理モデルパラメータ展開処理のフローチャートを示す図である。

【図14】 本発明における物理モデル音源の波形発生処理のフローチャートの一部を示す図である。

【図15】 本発明における物理モデル音源の波形発生処理のフローチャートの残る一部を示す図である。

【図16】 波形発生処理中の物理モデル演算処理のフローチャートを示す図である。

【図17】 物理モデル演算処理中の遅延ループ部演算処理のフローチャートを示す図である。

【図18】 モデル音源における遅延回路の遅延時間長を制御する方法を説明するための図である。

【図19】 モデル音源における単位遅延手段の遅延時間長を制御する方法を説明するための図である。

【図20】 各音色の音制御パラメータVATONEPARが格納されている状態の例を示す図である。

【図21】 本発明に係る物理モデル音源における遅延回路の構成をハードウェア的に示す図である。

【図22】 図21に示す遅延回路における動作モードを説明するための図である。

【図23】 本発明に係る物理モデル音源における遅延回路における遅延領域の割り当て動作を説明するための図である。

【図24】 本発明に係る複数の発音チャンネルを有する物理モデル音源における遅延回路における遅延領域の割り当て動作を説明するための図である。

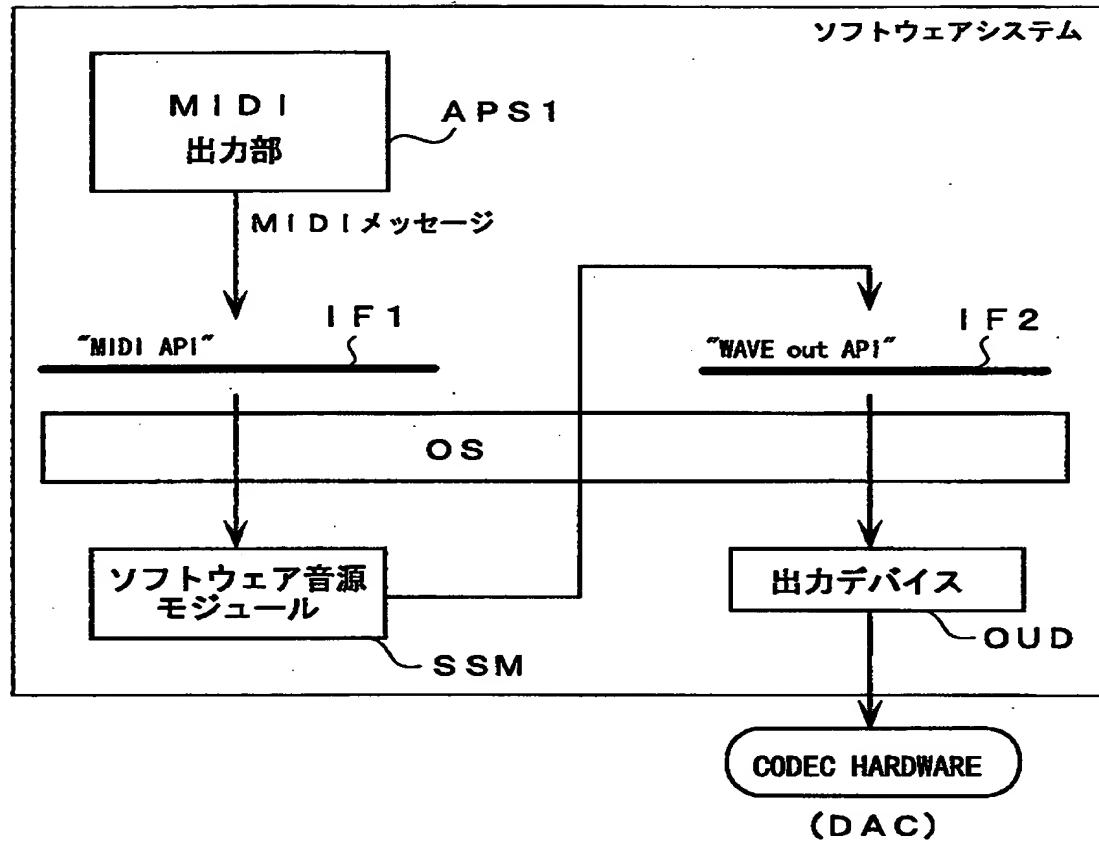
【符号の説明】

1 CPU、2 ROM、3 RAM、4 ディスプレイインターフェース、5
ディスプレイ、6 ハード・ディスク装置、7 CD-ROM装置、7-1
CD-ROM、8 インターフェース、9 音源ボード、9-1 DSP、9-
2 RAM、10 キーボード、11 ネットワークインターフェース、12
MIDIインターフェース、13 演奏操作子、14 CODEC、14-1 サンプル
バッファ、14-2 DAC、20, 21 RAM上に確保された遅延領域、3
1, 32, 33 セレクタ、34 合成部

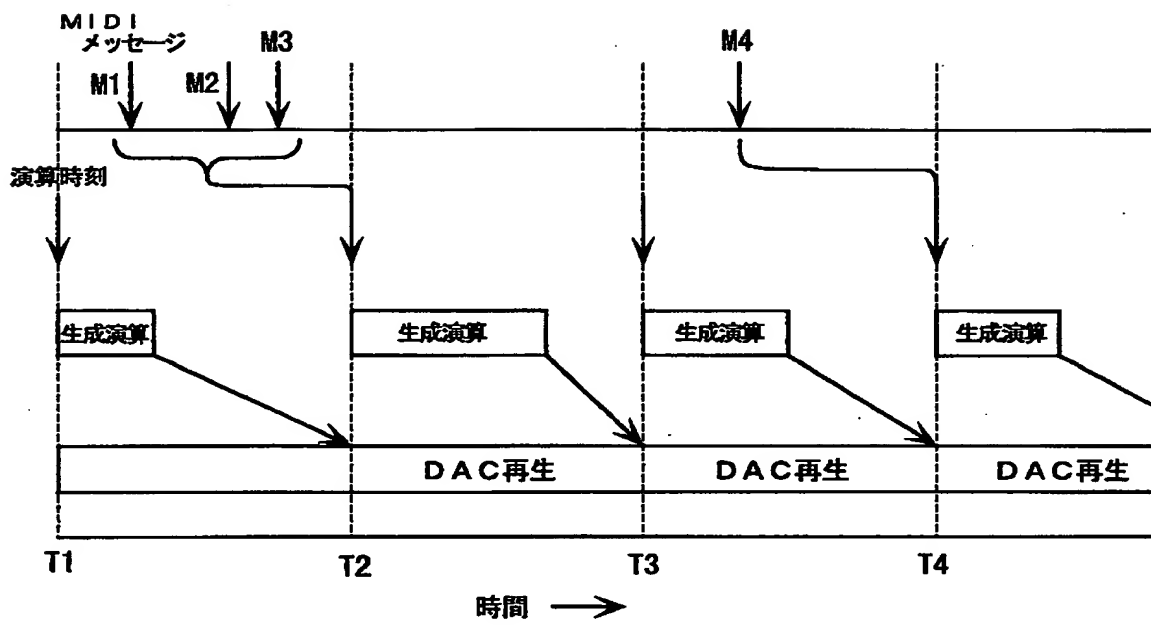
【書類名】

図面

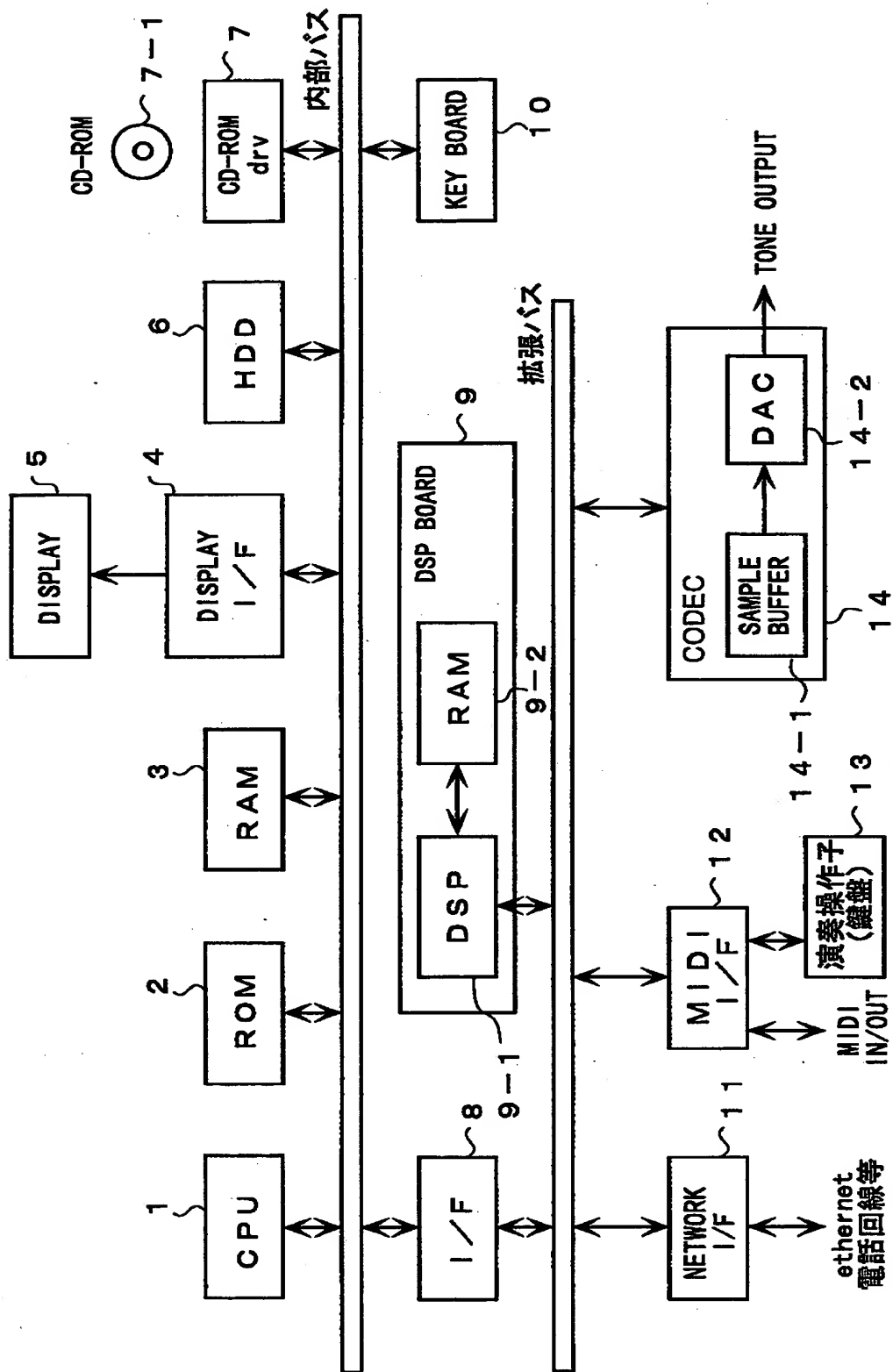
【図1】



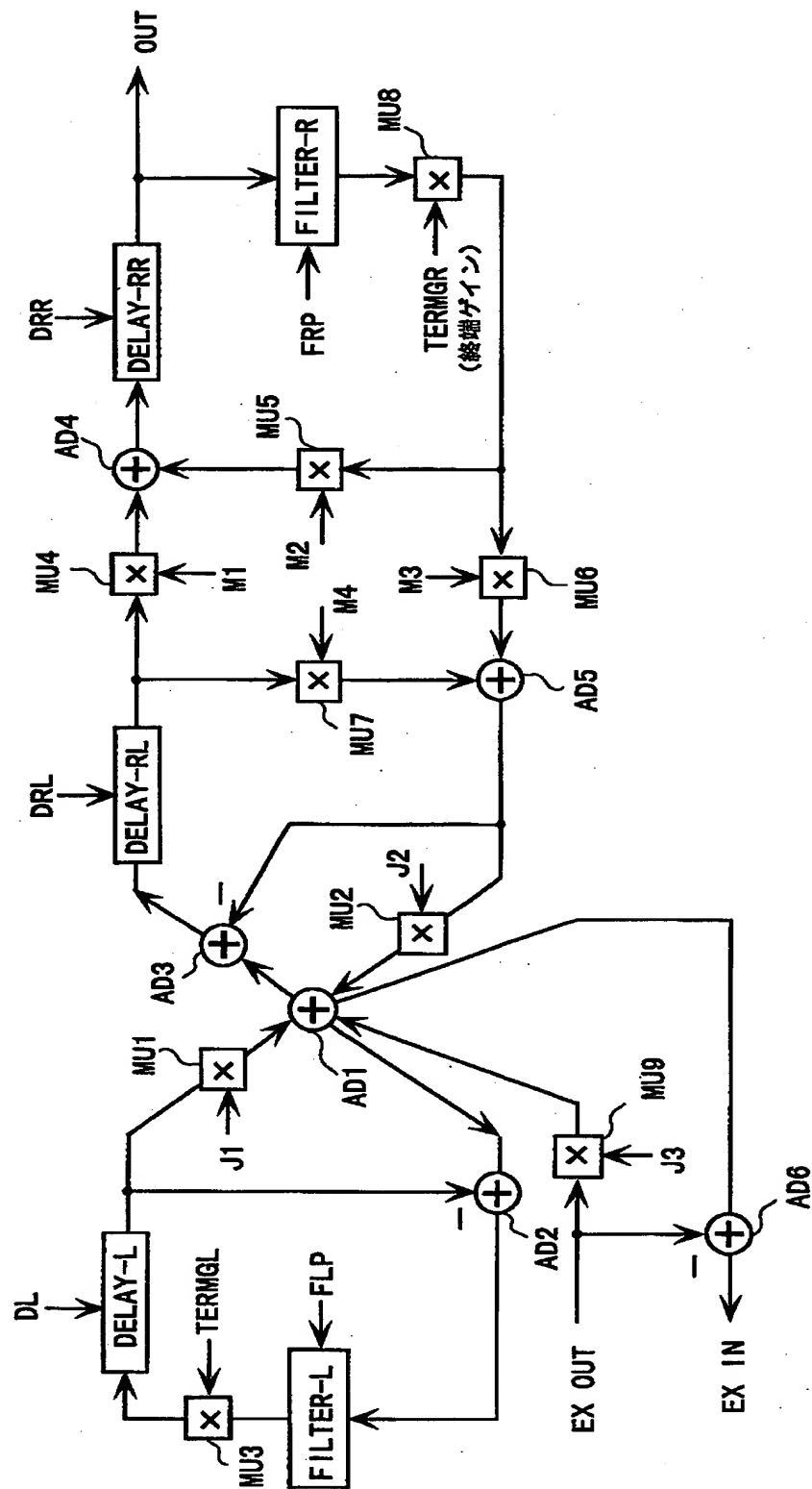
【図2】



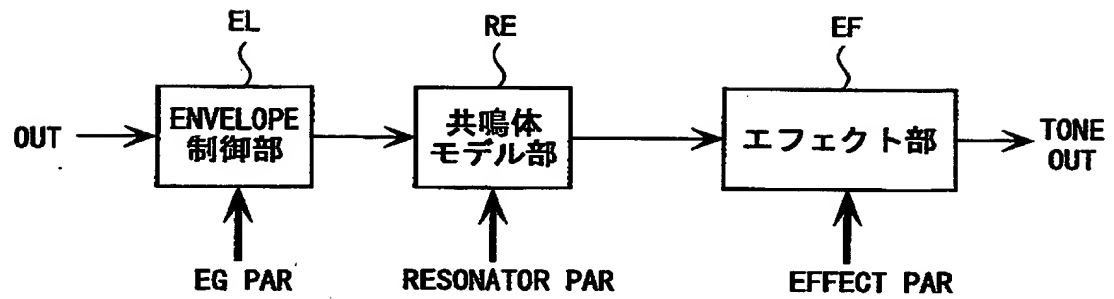
【図3】



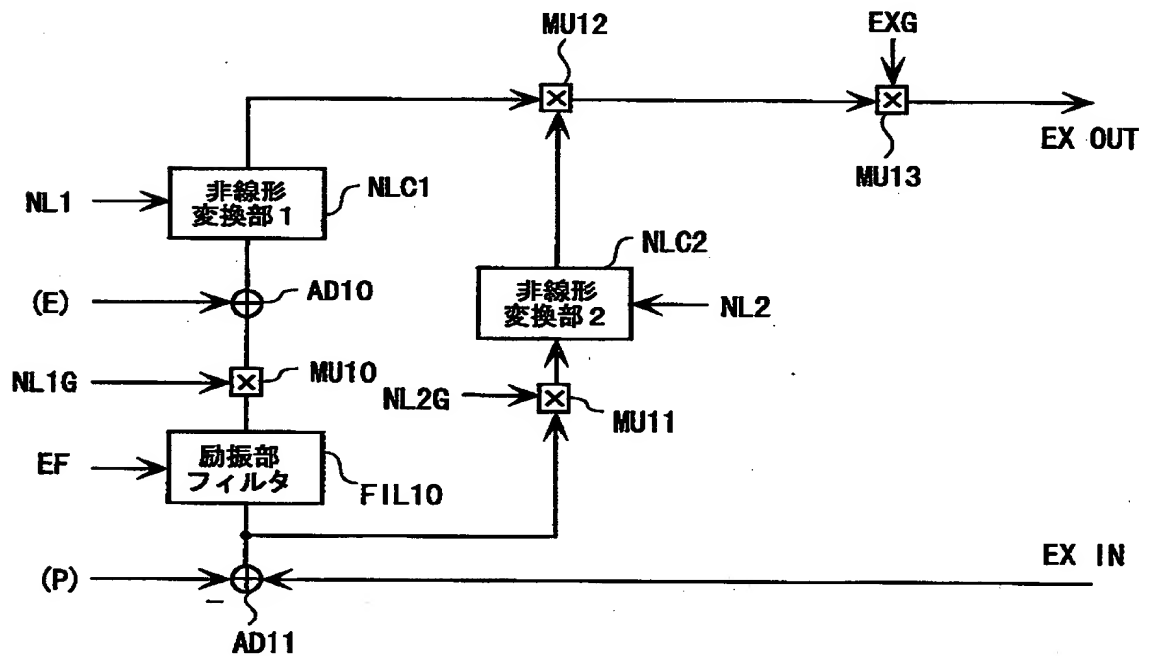
【図4】



【図 5】



【図 6】



【図 7】

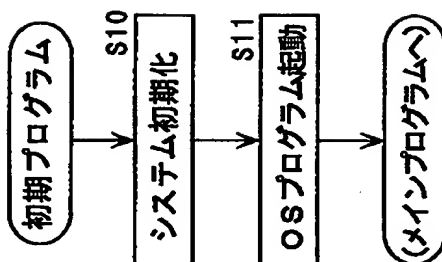


【図 8】

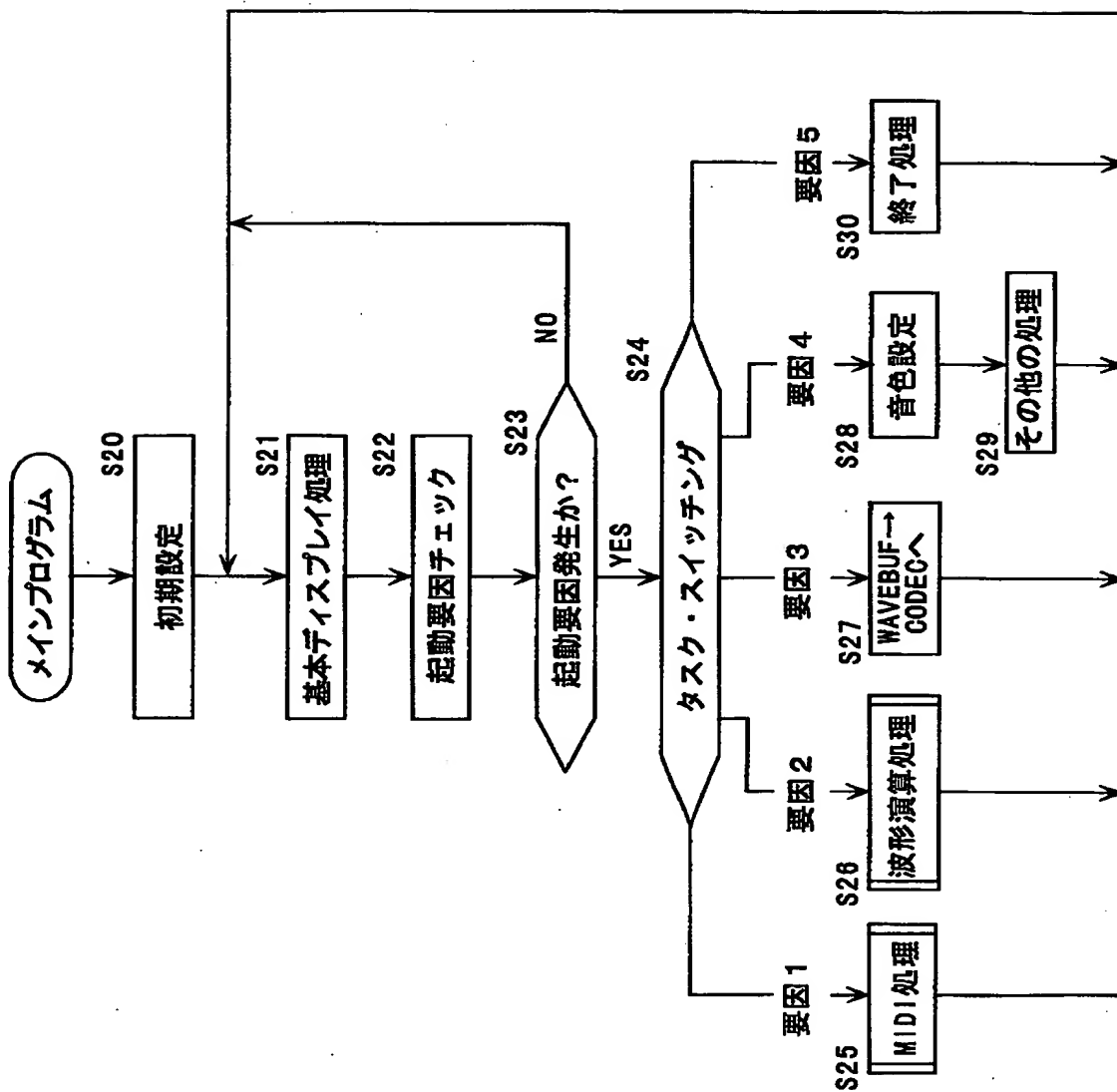
VATONEPAR

EXICITER PARAMETERS	EF	励振部フィルタパラメータ
	NLG1	非線形変換部1入力ゲイン
	NLG2	非線形変換部2入力ゲイン
	EXG	励振部出力ゲイン
	NL1	非線形変換部1特性パラメータ (テーブル)
	NL2	非線形変換部2特性パラメータ (テーブル)
P/S PARAMETERS	DL	DELAY-L 遅延量テーブル
	DRL	DELAY-RL 遅延量テーブル
	DRR	DELAY-RR 遅延量テーブル
	FLP	終端 FILTER-L パラメータ
	FRP	終端 FILTER-R パラメータ
	MULTI1 (M1)	トーンホールジャンクション乗算係数 1
	MULTI2 (M2)	トーンホールジャンクション乗算係数 2
	MULTI3 (M3)	トーンホールジャンクション乗算係数 3
	MULTI4 (M4)	トーンホールジャンクション乗算係数 4
	J1	管ジャンクション乗算係数 1
	J2	管ジャンクション乗算係数 2
	J3	管ジャンクション乗算係数 3
EG PAR	ATTACK RATE	アタックレート
	RELEASE RATE	リリースレート
	↓	↓
RESONATOR PAR	TYPE	共鳴器タイプ
	FREQ	共鳴器周波数特性パラメータ
	LEVEL	共鳴器レベルパラメータ
	↓	↓
EFFECT PAR	EFFECT TYPE	エフェクトタイプ指定
	FREQ	エフェクト深さ
	LEVEL	変調スピード
	↓	↓
SAMPLING FREQ.	FS	サンプリング周波数データ (FS1>FS2)

【図9】

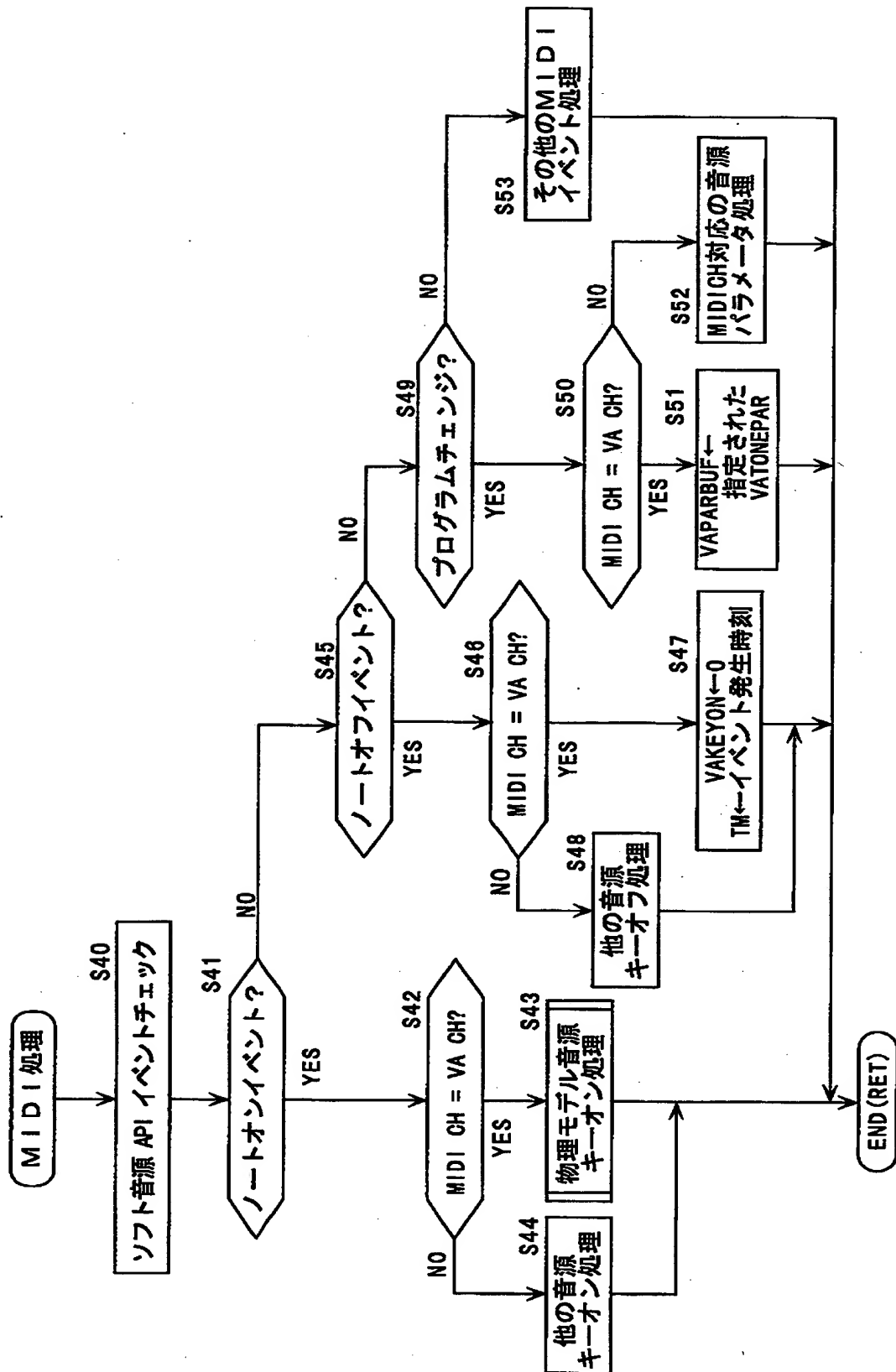


【図10】

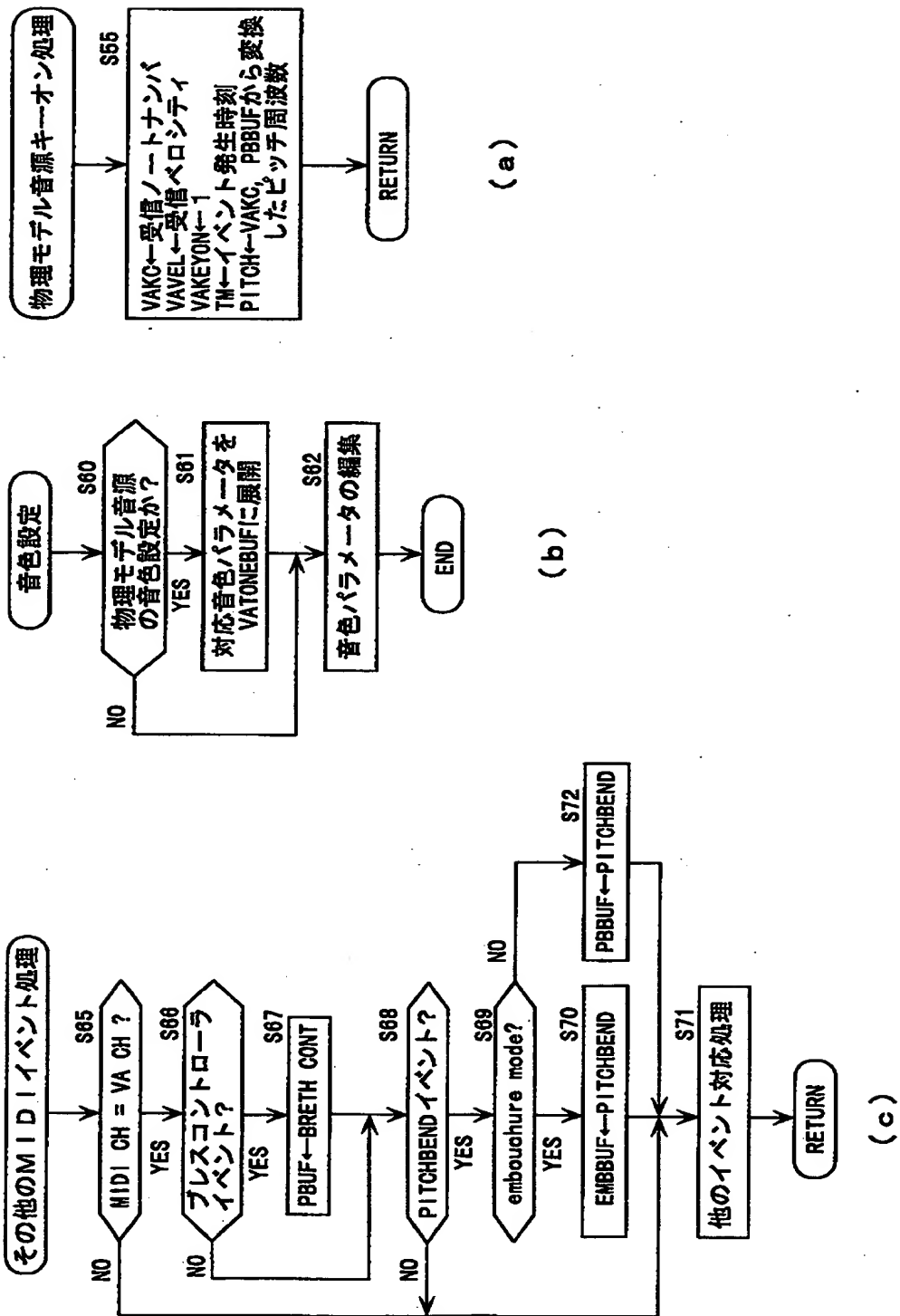


特平 9-017333

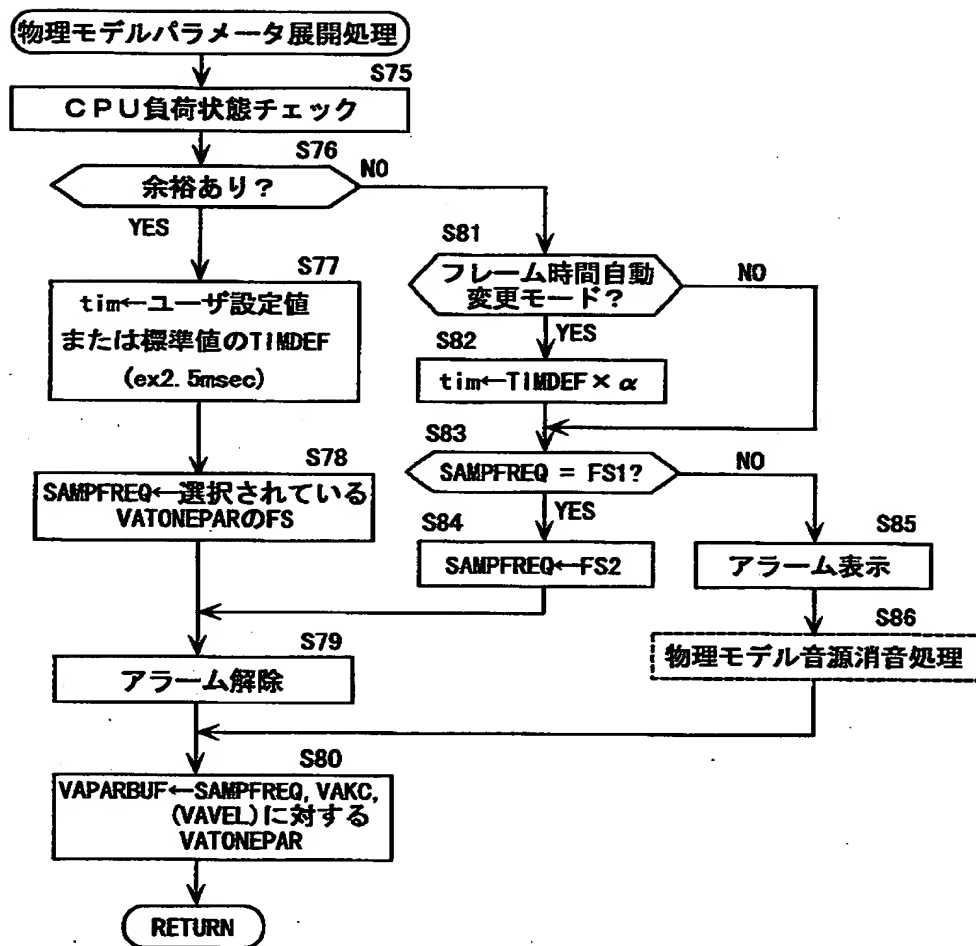
【図11】



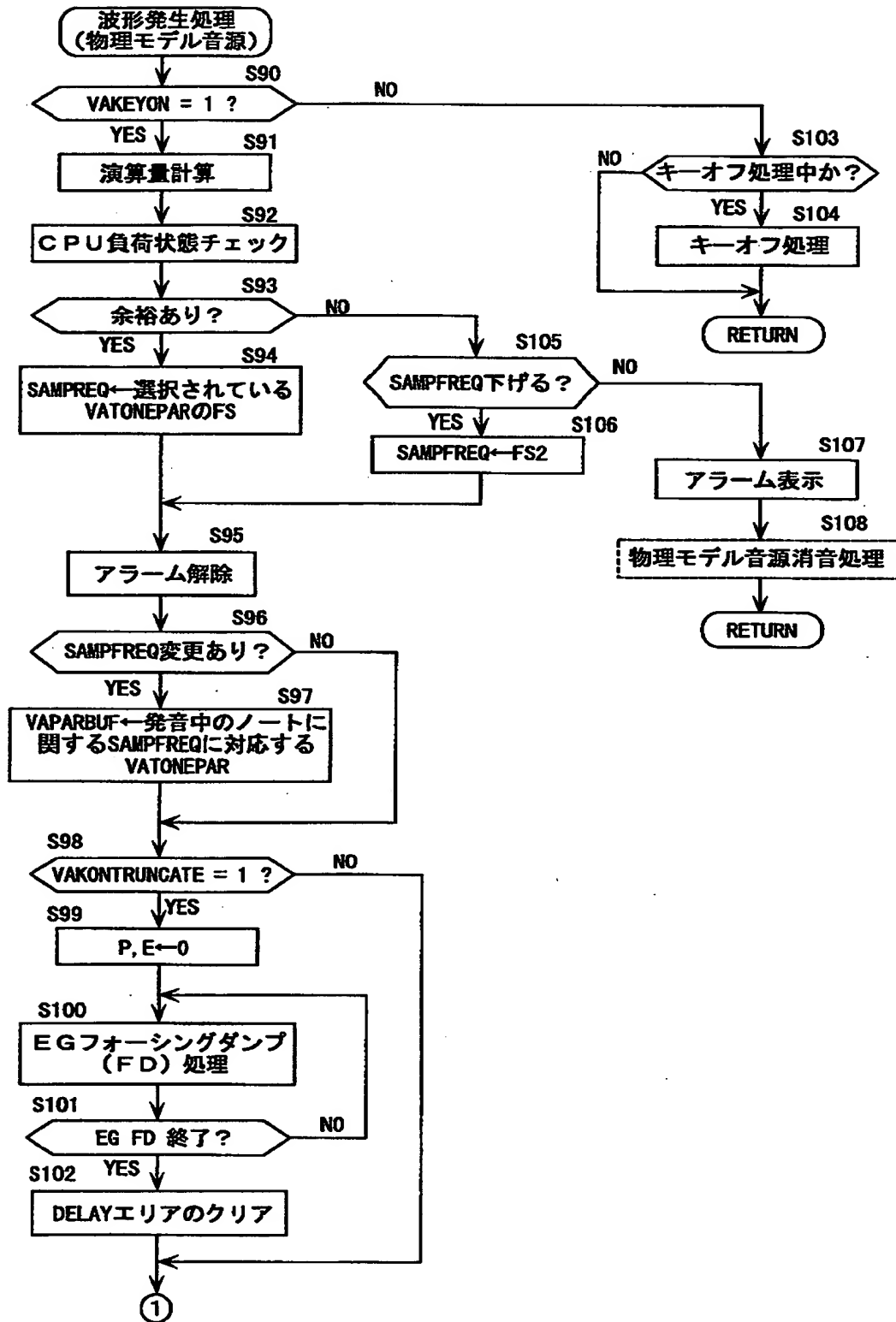
【図12】



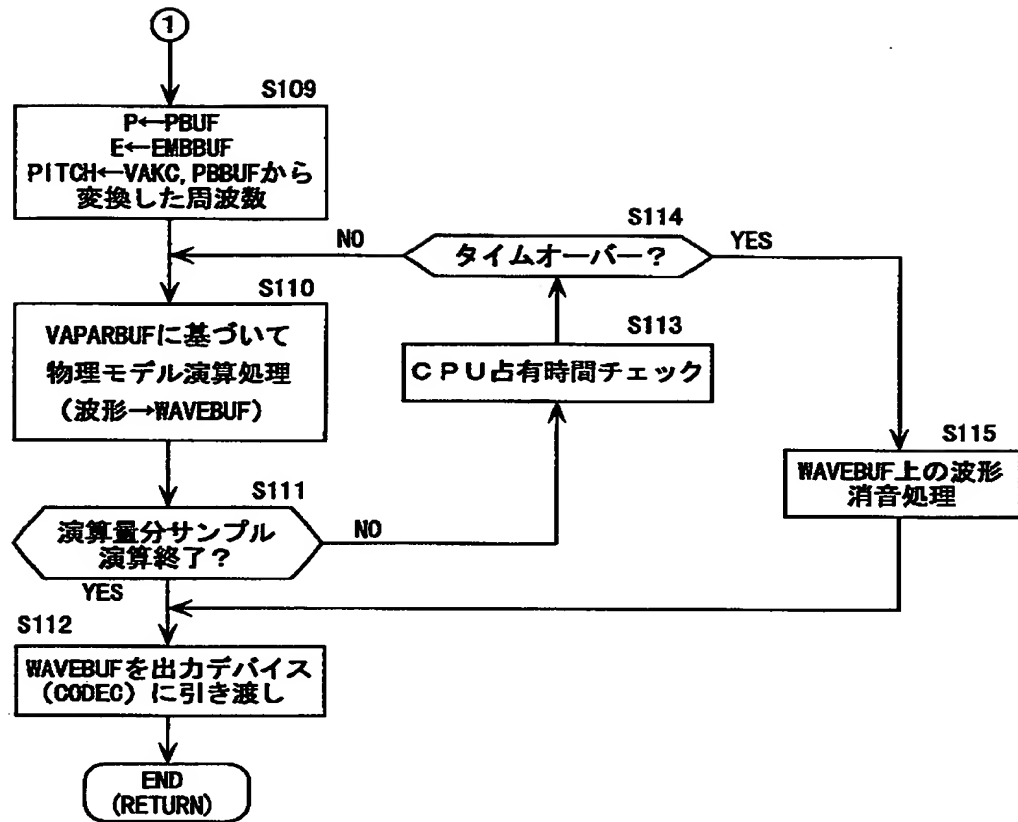
【図13】



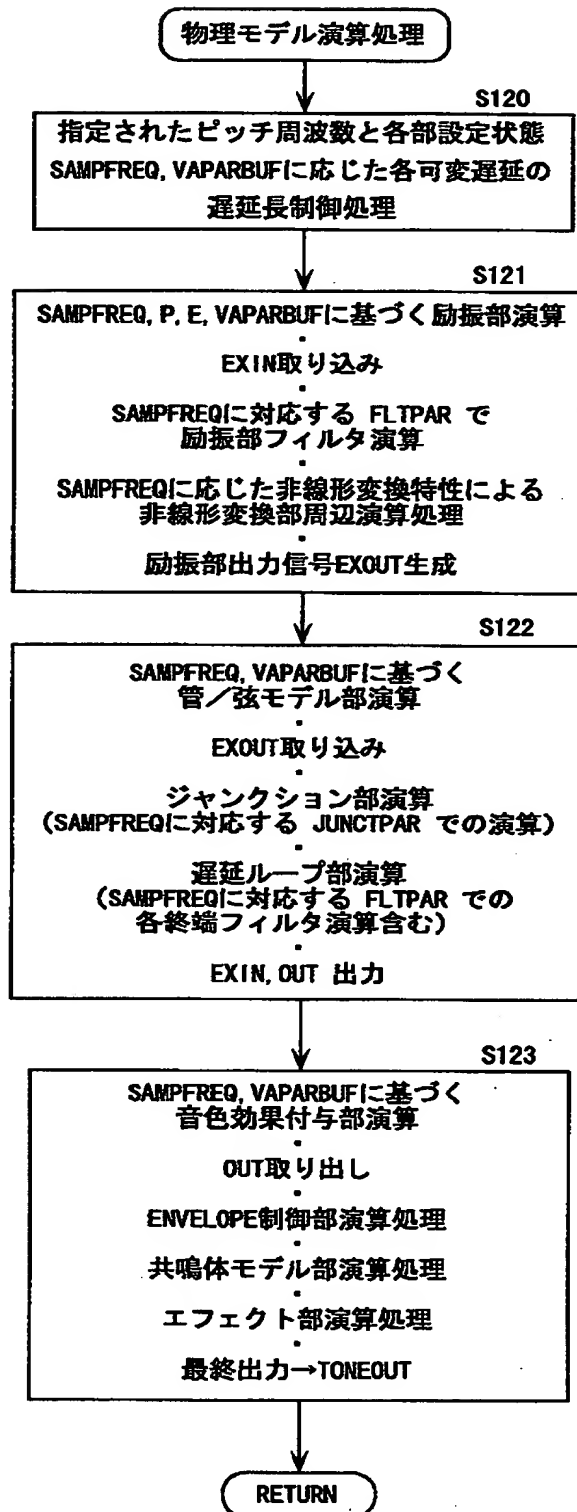
【図14】



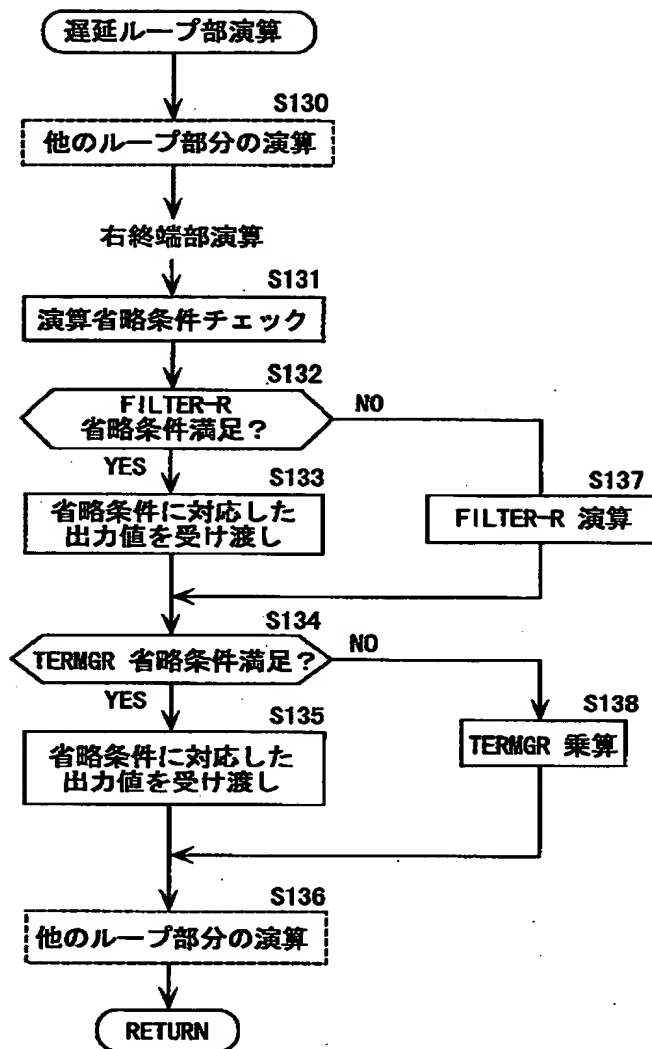
【図15】



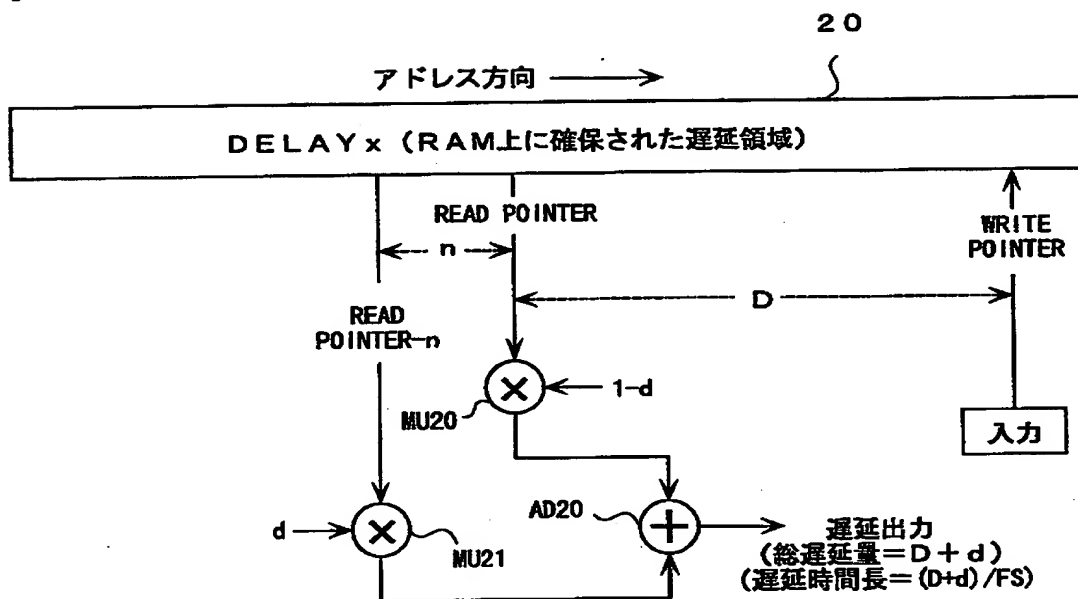
【図16】



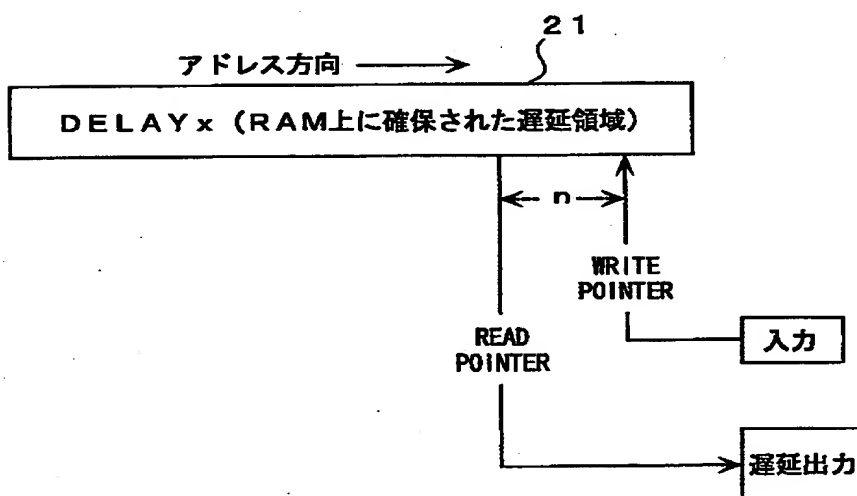
【図17】



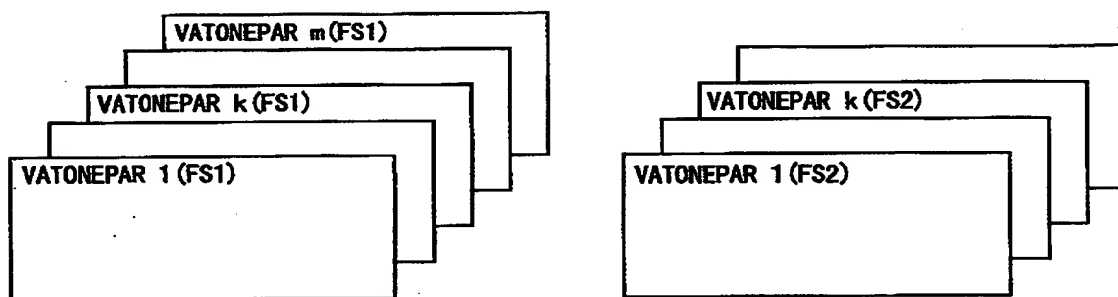
【図18】



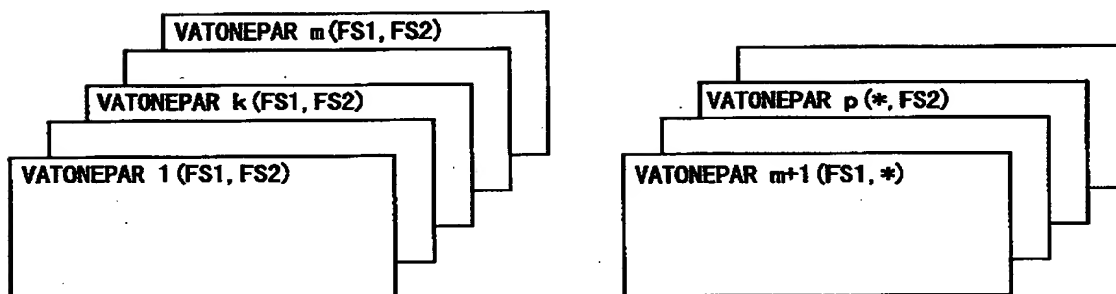
【図19】



【図20】

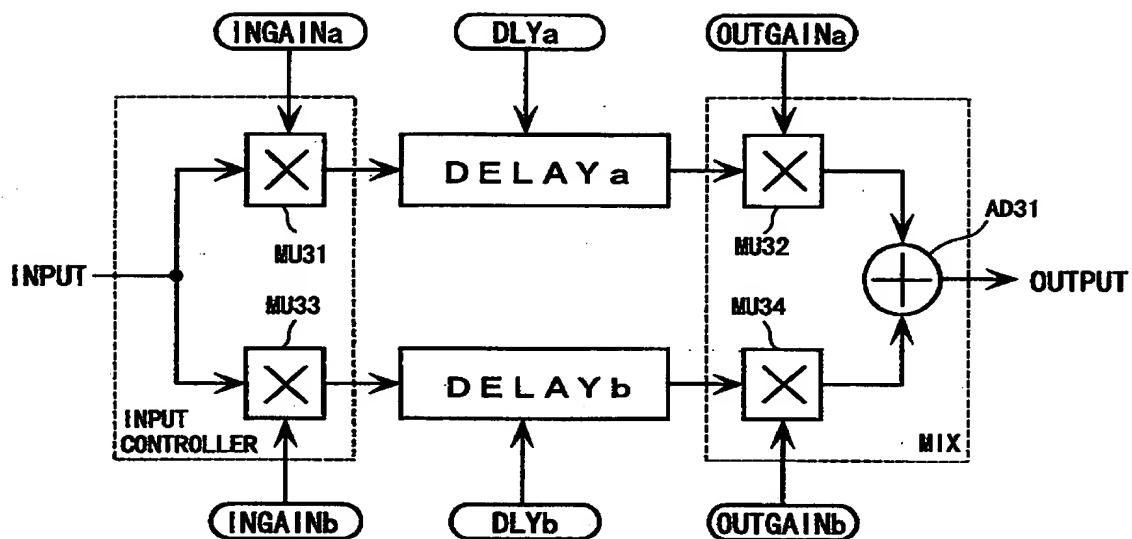


(a)

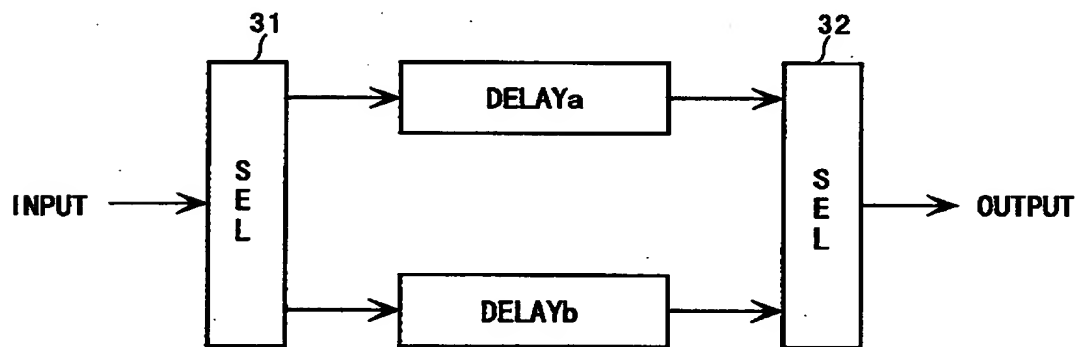


(b)

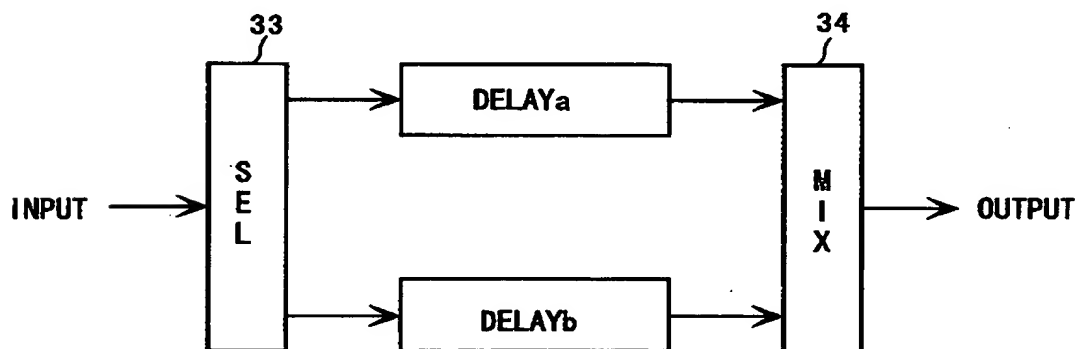
【図 2 1】



【図 2 2】

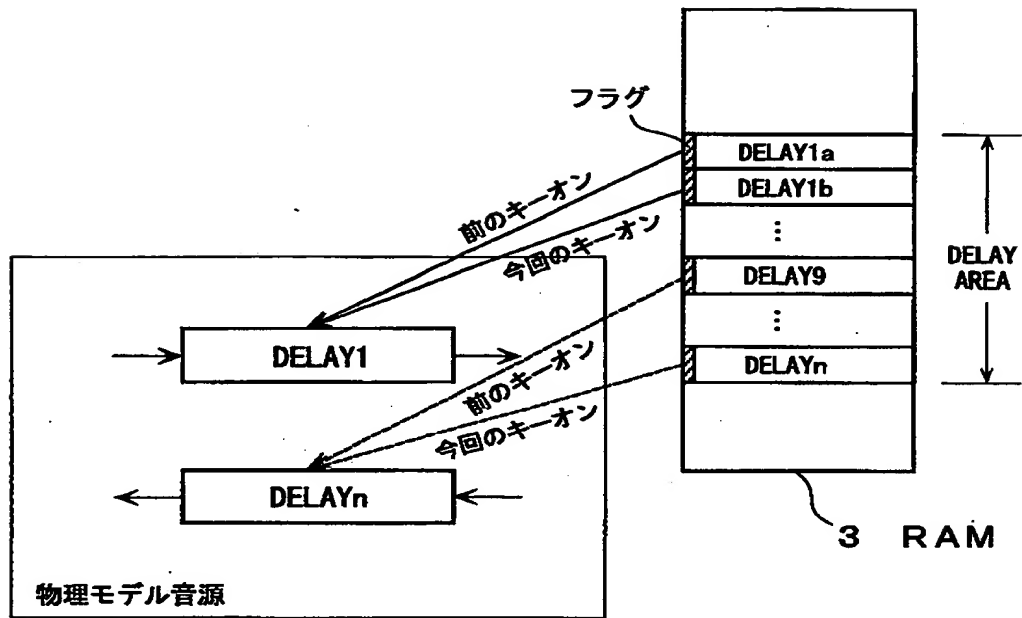


(a)

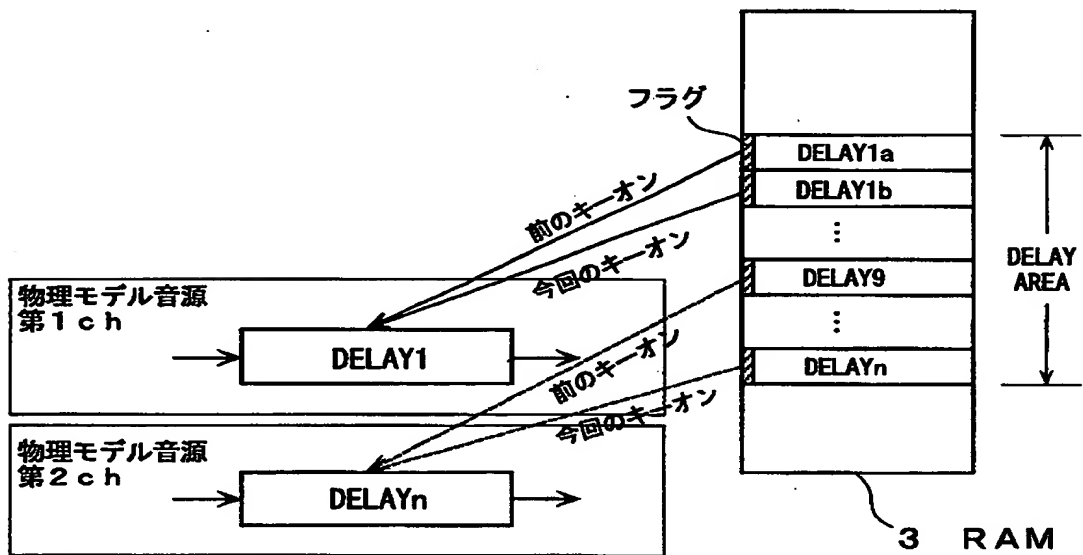


(b)

【図 2 3】



【図 2 4】



【書類名】 要約書

【要約】

【課題】 演算生成による波形サンプルデータが途切れないようにする。

【解決手段】 CPUはフレーム周期の演算時刻ごとに生成演算を行って、継続音および直前のフレームで受信されたMIDIメッセージに基づく新たな音の波形サンプルを生成する。このときCPUの負荷をチェックして、余裕がないときは波形サンプルのサンプリング周期を $1/n$ に変更して必要とする波形サンプル数を低減する。また、実質的にゲインがゼロとされている部分の演算を省略して、CPUの負荷を軽くする。

【選択図】 図2

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【手数料の表示】

【納付金額】 0円

【特許出願人】

【識別番号】 000004075

【住所又は居所】 静岡県浜松市中沢町10番1号

【氏名又は名称】 ヤマハ株式会社

【代理人】

【識別番号】 100102635

【住所又は居所】 東京都中央区新川1丁目27番8号 新川大原ビル
6階 雄渾特許事務所

【氏名又は名称】 浅見 保男

【代理人】

【識別番号】 100106459

【住所又は居所】 東京都中央区新川1丁目27番8号 新川大原ビル
6階 雄渾特許事務所

【氏名又は名称】 高橋 英生

【代理人】 申請人

【識別番号】 100086841

【住所又は居所】 東京都中央区新川1丁目27番8号 新川大原ビル
6階

【氏名又は名称】 脇 篤夫

【書類名】 手続補正書

【提出日】 平成 9年 3月27日

【あて先】 特許庁長官殿

【事件の表示】

 【出願番号】 平成 9年特許願第 17333号

【発明の名称】 音源波形データ生成方法および装置

【補正をする者】

 【事件との関係】 特許出願人

 【識別番号】 000004075

 【氏名又は名称】 ヤマハ株式会社

【代理人】

 【識別番号】 100102635

 【弁理士】

 【氏名又は名称】 浅見 保男

【代理人】

 【識別番号】 100106459

 【弁理士】

 【氏名又は名称】 高橋 英生

【発送番号】 007916

【手続補正 1】

 【補正対象書類名】 特許願

 【補正対象項目名】 代理人

 【補正方法】 追加

 【補正の内容】

 【その他】 本件特許出願の手続をしたことに相違ありません。

【手数料補正】

 【補正対象書類名】 特許願

 【予納台帳番号】 037338

 【納付金額】 21,000円

【書類名】 職権訂正データ
【訂正書類】 手続補正書

<認定情報・付加情報>

【補正をする者】

【識別番号】 000004075

【住所又は居所】 静岡県浜松市中沢町10番1号

【氏名又は名称】 ヤマハ株式会社

【代理人】 申請人

【識別番号】 100102635

【住所又は居所】 東京都中央区新川1丁目27番8号 新川大原ビル
6階 雄渾特許事務所

【氏名又は名称】 浅見 保男

【代理人】 申請人

【識別番号】 100106459

【住所又は居所】 東京都中央区新川1丁目27番8号 新川大原ビル
6階 雄渾特許事務所

【氏名又は名称】 高橋 英生

出 願 人 履 歴 情 報

識別番号 [000004075]

1. 変更年月日 1990年 8月22日

[変更理由] 新規登録

住 所 静岡県浜松市中沢町10番1号

氏 名 ヤマハ株式会社